

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PROJETO ASSISTIDO POR COMPUTADOR PARA SISTEMAS  
DE CONTROLE: ESPECIFICAÇÃO E PROJETO DE UM  
PACOTE

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA

VÂNIO MURILO SAVI

FLORIANÓPOLIS, MAIO - 1987

PROJETO ASSISTIDO POR COMPUTADOR PARA SISTEMAS DE CONTROLE:

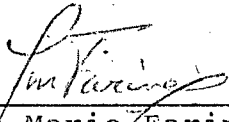
ESPECIFICAÇÃO E PROJETO DE UM PACOTE

VÂNIO MURILO SAVI

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO  
TÍTULO DE

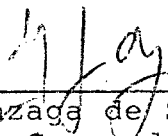
MESTRE EM ENGENHARIA ELÉTRICA

ESPECIALIDADE ENGENHARIA ELÉTRICA E APROVADA EM SUA FORMA FINAL  
PELO PROGRAMA DE PÓS-GRADUAÇÃO



---


Prof. Jean-Marie Farines  
Orientador



---


Prof. Luiz Gonzaga de Souza Fonseca  
Coordenador do Curso de Pós-graduação  
em Engenharia Elétrica

BANCA EXAMINADORA



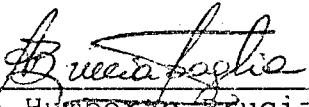
---

Prof. Jean-Marie Farines, Dr. Ing.  
Presidente



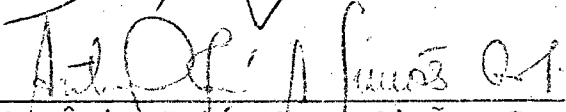
---

Prof. Fernando Antônio Campos Gomide, Ph.D.



---

Prof. Augusto Humberto Bruciapaglia, Dr. Ing.



---

Prof. Antônio José Alves Simões Costa, Ph. D.

À minha família.

## AGRADECIMENTOS

Ao prof. Jean-Marie Farines pela orientação e, principalmente, pela dedicação e amizade demonstradas no decorrer do desenvolvimento deste trabalho.

A todos os os professores, colegas e funcionários' do programa de pós-graduação do Departamento de Engenharia Elétrica que, de uma forma ou de outra, contribuíram para a realização deste trabalho. Especialmente ao colega Eugênio de Bona Castelan Neto e ao prof. Augusto Humberto Bruciapaglia pelas sugestões apresentadas.

A minha irmã Monica Rosane Savi pelos desenhos.

A UFSC e ao CNPQ pelo apoio financeiro que permitiram a realização deste trabalho.

## S U M Á R I O

RESUMO .....	ix
ABSTRACTS.....	x
INTRODUÇÃO.....	1
CAPÍTULO 1 - INTRODUÇÃO.....	2
PARTE 1 - ENGENHARIA DE CONTROLE ASSISTIDA POR COMPUTADOR..	4
CAPÍTULO 2 - ENGENHARIA DE CONTROLE ASSISTIDA POR COMPUTADOR.....	5
2.1 - Projeto.....	5
2.2 - Projeto assistido por computador.....	6
2.3 - Projeto Assistido por Computador para Sistemas de Controle (PACSC).....	11
2.3.1 - Projeto em sistemas de controle: alguns princípios.....	11
2.3.2 - Evolução da engenharia de controle.....	13
2.3.3 - Descrição do estado da arte em PACSC.....	15
2.3.3.1 - Primeira geração de programas.....	15
2.3.3.2 - Segunda geração de programas.....	16
2.3.3.3 - Terceira geração de programas.....	22
2.4 - Os programas de PACSC no ensino da engenharia de controle.....	23
2.5 - A evolução dos programas de PACSC no LCMI.....	25
2.5.1 - Programas desenvolvidos no LCMI.....	25
2.5.2 - O programa proposto.....	26
2.6 - Conclusão.....	29

## PARTE 2 - UM PROGRAMA DE PROJETO ASSISTIDO POR COMPUTADOR

PARA SISTEMAS DE CONTROLE.....	30
CAPÍTULO 3 - METODOLOGIA ADOTADA.....	31
3.1 - Introdução.....	31
3.2 - Metodologia de desenvolvimento adotada.....	32
3.2.1 - Detalhes sobre a metodologia.....	33
3.2.2 - sistemática de trabalho.....	35
3.3 - Conclusão.....	36
CAPÍTULO 4 - DEFINIÇÃO DO PROGRAMA.....	37
4.1 - Introdução.....	37
4.2 - Objetivo.....	37
4.3 - Descrição geral do programa.....	37
4.4 - Descrição geral da parte interativa do programa.	39
4.4.1 - Introdução do objeto a tratar: declaração do sistema.....	39
4.4.2 - Escolha das ferramentas.....	40
4.4.3 - Visualização dos resultados.....	42
4.5 - Descrição detalhada da parte interativa do pro- grama: interface homem-máquina.....	44
4.5.1 - Janelas de comunicação.....	44
4.5.2 - Mecanismos da comunicação homem-máquina..	45
4.6 - Conclusão.....	53
CAPÍTULO 5 - ESTRUTURA GERAL DO PROGRAMA.....	54
5.1 - Descrição informal.....	54
5.1.1 - Gerenciador.....	54
5.1.2 - Unidades funcionais.....	56
5.1.3 - Interface homem-máquina.....	58

5.2 - Descrição formal.....	58
5.2.1 - Objetivo da descrição formal.....	58
5.2.2 - Metodologia de descrição formal.....	59
5.3 - Conclusão.....	79
CAPÍTULO 6 - ESTRUTURA DE DADOS.....	80
6.1 - Introdução.....	80
6.2 - Objetivo de uma estrutura para o armazenamento dos modelos do sistema.....	81
6.3 - Estrutura escolhida.....	82
6.4 - Descrição da estrutura de dados.....	83
6.4.1 - Características construtivas.....	83
6.4.2 - Montagem da árvore binária.....	87
6.4.3 - Tipos de estruturas de dados criadas.....	91
6.5 - Modificação da estrutura de dados do sistema....	92
6.6 - Acesso aos elementos da estrutura de dados.....	94
6.6.1 - Procura de elementos.....	97
6.6.2 - Recuperação dos dados armazenados.....	98
6.6.3 - Recuperação da equações.....	99
6.7 - Conclusão.....	100
CAPÍTULO 7 - DESCRIÇÃO DAS FERRAMENTAS DO PROGRAMA....	101
7.1 - Redução formal.....	101
7.1.1 - Introdução.....	101
7.1.2 - O método de redução formal.....	102
7.2 - Transformação de modelos.....	107
7.3 - Ferramentas clássicas....	107
7.3.1 - Lugar das raízes.....	108
7.3.2 - Resposta frequencial.....	108
7.4 - Ferramentas modernas.....	109

7.4.1 - Definição da estrutura da servocompensa-	
dor.....	112
7.4.2 - Obtenção das matrizes do servocompensa-	
dor.....	114
7.5 - Simulação.....	119
7.6 - Conclusão.....	121
CONCLUSÃO.....	122
CAPÍTULO 8 - CONCLUSÃO.....	123
ANEXOS.....	126
ANEXO 1 - STRUCTURED ANALYSIS AND DESIGN TECHNIQUE:	
SADT.....	127
ANEXO 2 - PRINCIPAIS TIPOS DE ESTRUTURAS DE DADOS.....	133
BIBLIOGRAFIA.....	138



## R E S U M O

Neste trabalho, são apresentados aspectos relativos à concepção de um programa interativo para análise, projeto e simulação de sistemas de controle monovariáveis ou multivariáveis, invariantes no tempo, contínuos ou discretos, em fase de implementação num microcomputador de 16 bits compatível IBM-PC-XT.

O programa é subdividido em três partes principais:

- um gerenciador que sequencia as operações do programa, ativando as tarefas a serem realizadas;
- um conjunto de unidades funcionais que contém as ferramentas de análise e projeto e as unidades de coordenação de arquivos, de tratamento de erros, de análise de dados e de tratamento gráfico;
- uma interface homem-máquina oferecendo facilidades tais como: janelas de interação, menus de seleção e de auxílio, diálogos do tipo pergunta-resposta para a entrada de dados e diferentes possibilidades de visualização dos resultados e de manipulações gráficas.

Apresenta-se também a base de dados do programa, destacando as estruturas para armazenamento dos resultados, dos dados de diálogos e dos modelos dos sistemas. As operações de montagem, modificação e acesso da estrutura que armazena o modelo do sistema são descritas em detalhes.

## ABSTRACT

This work presents some aspects related to the conception of an interactive program for analysis, design and simulation of mono/multivariable, time invariant, continuous/discrete control systems. The program has been implementing in a 16 bits IBM-PC-XT compatible microcomputer.

This program is subdivided in the three following parts:

- a supervisor that schedules the program operation activating tasks;
- an ensemble of functional units including analysis and design tools, file handler, errors treatment, data analysis and graphic treatment;
- a man-machine interface allowing facilities as interaction windows, help and selection menus, questions and answers dialogues, and different possibilities of results visualization and graphics manipulation.

It is also described the program's data base, emphasizing results, dialogues and system models storage. The operations involved in assembling, modifying and accessing the system models structure are presented in details.

## INTRODUÇÃO

## C A P Í T U L O 1

### INTRODUÇÃO

A evolução da tecnologia com o conseqüente aumento das potencialidades computacionais em termos da representação gráfica da informação, da rapidez do tratamento desta e da capacidade de compactação no seu armazenamento, propiciou o surgimento de uma nova abordagem para a realização do projeto nas várias áreas da ciência, em particular na engenharia. Esta nova abordagem, baseada no uso efetivo do computador no projeto, surgiu nos países desenvolvidos no decorrer dos anos 70 atingindo nestes um alto grau de desenvolvimento, como pode ser verificado pelo grande número de pacotes de programas computacionais desenvolvidos nas mais variadas áreas da engenharia; esses programas não se limitaram ao meio acadêmico, mas têm amplo uso no meio industrial. Entretanto, em países em desenvolvimento, como o Brasil, não se verificou uma evolução ao mesmo nível por razões óbvias de falta de recursos materiais e humanos, estabelecendo, como conseqüência, um atraso substancial nesta área.

A introdução no país dessas importantes ferramentas de apoio ao projeto, implica na necessidade de se realizar um esforço no sentido de adquirir, o mais rapidamente possível, uma competência

tanto na utilização dos programas como na sua produção. Com o objetivo de contribuir para este esforço na área de engenharia de controle, o Laboratório de Controle e Microinformática do Departamento de Engenharia Elétrica da UFSC (LCMI-EEL-UFSC) está desenvolvendo programas utilizáveis tanto a nível industrial como acadêmico.

Esta dissertação tem como objetivo apresentar as especificações completas e elementos do projeto de um programa de Projeto Assistido por Computador para Sistemas de Controle (PACSC). Apesar deste programa ainda estar em fase de implementação, considerar-se-á, com o objetivo de facilitar a sua descrição, este como implementado.

A apresentação do trabalho está dividida em três partes: a primeira parte apresenta a evolução da engenharia de controle assistida por computador e, dentro deste contexto, os desenvolvimentos efetuados no LCMI nesta área; a segunda parte apresenta as especificações do programa, divididas em capítulos que tratam da sua definição, da sua estrutura geral, da estrutura de dados escolhida e da descrição das ferramentas a serem utilizadas no programa; a terceira parte apresenta as conclusões do trabalho, as perspectivas de implementação e propostas para futuros trabalhos na área.

P A R T E 1

ENGENHARIA DE CONTROLE ASSISTIDA POR COMPUTADOR

## C A P Í T U L O 2

### ENGENHARIA DE CONTROLE ASSISTIDA POR COMPUTADOR

#### 2.1 - Projeto

O projeto em engenharia pode ser definido como o conjunto de atividades que permitem propor uma alternativa concreta e viável de solução para um problema cujos objetivos foram pré-estabelecidos; esta fase antecede sempre a fase de implementação da solução (construção de um sistema, fabricação de um produto, etc.). O projeto necessita de um conhecimento profundo dos objetivos a serem alcançados e da sua análise em função de experiências anteriores, e das técnicas e ferramentas existentes; a partir disto, poderá se realizar um processo de tomada de decisão, o qual levará a escolha da alternativa mais adequada entre aquelas oriundas da criatividade do projetista.

A metodologia básica de projeto tem como ponto de partida um conjunto de especificações anteriormente validadas, e que definam o objeto (ou sistema) a ser realizado em termos de funções. Os requisitos de confiabilidade, de apresentação, de facilidades de uso, de manutenção, de expansão, e as restrições de custo e tecnológicas devem também ser consideradas nesta fase.

Em seguida, formulam-se alternativas cada vez mais refinadas

e precisas, visando a realização do objeto (ou sistema) desejado; estas alternativas são representáveis por modelos, a partir dos quais poderão ser observados os respectivos comportamentos e a sua adequação às especificações anteriormente fixadas, num processo chamado de análise. A avaliação destas observações orientará o processo de escolha da melhor alternativa a ser implementada (ou fabricada) numa fase posterior (TOZZI, 1986).

## 2.2 - Projeto assistido por computador

A introdução dos computadores no projeto deu origem a novas concepções e técnicas que podem ser agrupadas no que se convencionou chamar de Projeto Assistido por Computador (PAC), ou Computer Aided Design (CAD). O uso dos computadores permitiu, num primeiro tempo, colocar à disposição do projetista um conjunto de ferramentas matemáticas orientadas para o projeto de determinado objeto e meios para armazenar os resultados de projetos anteriores, permitindo desta forma a eliminação de tarefas repetitivas e tediosas de cálculo, e o fácil acesso a um registro de informações de projeto. Outrossim, graças aos avanços tecnológicos alcançados (microcomputadores, periféricos gráficos, etc.), as técnicas de PAC evoluíram no sentido de facilitar a visualização dos resultados obtidos e na utilização do desenho como forma de comunicação homem-máquina. Enfim, tornou-se também possível o auxílio ao projetista no processo de tomada de decisões e de escolha de alternativas, a partir do acesso a representações do conhecimento conceitual de especialistas (ESMANHOTO, 1983).

A utilização dos computadores em projeto permite, então, que



a atenção do projetista seja dirigida prioritariamente para as atividades criativas e de tomada de decisões, deixando-se de se preocupar com atividades secundárias (cálculo, relatório, gráficos, etc.). Do uso das técnicas de PAC, resultam um aumento da qualidade e da confiabilidade dos projetos, uma redução do tempo e do custo destes, e um aumento de flexibilidade.

#### a - Características dos sistemas de PAC

Pelo fato da simples adaptação das técnicas convencionais não levarem a resultados satisfatórios, tornou-se necessário a elaboração de novos métodos de projeto baseados em formas de representação e comunicação mais adequadas. Por esta razão, todo sistema de projeto assistido por computador deve atender às seguintes características (TOZZI, 1985) :

- capacidade gráfica de entrada e saída;
- existência de uma linguagem formal de descrição e de acesso às ferramentas, orientada para o objeto de aplicação do sistema;
- existência de base de dados que será utilizada para troca de informações entre as várias partes do sistema de PAC;
- existência de uma base de ferramentas ou métodos atualizados;
- facilidade de interação do usuário com o sistema PAC através de uma interface homem-máquina adaptada ao problema.

Mais recentemente, começou a ser introduzida uma característica nova neste tipo de sistema, baseada na existência de uma base de conhecimentos contendo um conjunto de informações

não formalizáveis oriundas de experiências anteriores, as quais caracterizam uma perícia em termos de determinado tipo de projeto.

b - Áreas de aplicação dos sistemas de PAC

Os sistemas de Projeto Assistido por Computador estão sendo, hoje em dia, largamente utilizados em quase todas as áreas da engenharia. Dentre as aplicações onde o uso do computador está bastante difundido pode-se citar: projeto de circuitos integrados, projeto de circuitos impressos, projeto de leis de controle em sistemas realimentados, projeto de peças mecânicas, projeto de máquinas, cálculo de estruturas, cálculo de tubulações, projeto de motores elétricos, etc (TOZZI, 1986).

c - Requisitos materiais e de programação dos sistemas de PAC

Os sistemas de Projeto Assistido por Computador devem ter um suporte material e de programação adaptados ao tipo de aplicação para o qual ele deve ser utilizado. Em termos de material, as várias configurações possíveis deverão conter alguns dos dispositivos seguintes:

- dispositivos de processamento, os quais poderão ser, dependendo das necessidades associadas a cada tipo de projeto e das disponibilidades financeiras: "mainframes" ou minicomputadores com várias estações de trabalho, estações autônomas baseadas em microcomputadores do tipo pessoal, ou redes contendo estações de trabalho simples baseadas em microcomputadores, tendo acesso a capacidade de processamento e de armazenamento de computadores mais potentes;

- dispositivos de entrada, tais como: teclado alfanumérico, mesa digitalizadora, "light pen", "mouse", etc.
- dispositivos de saída, tais como: monitores gráficos, registradores X-Y, impressora gráfica matricial ou a laser, "hard-copy", etc.

Um exemplo de uma configuração típica de hardware para um sistema PAC de grande porte com processamento centralizado e estações de trabalho é apresentada na figura 1.

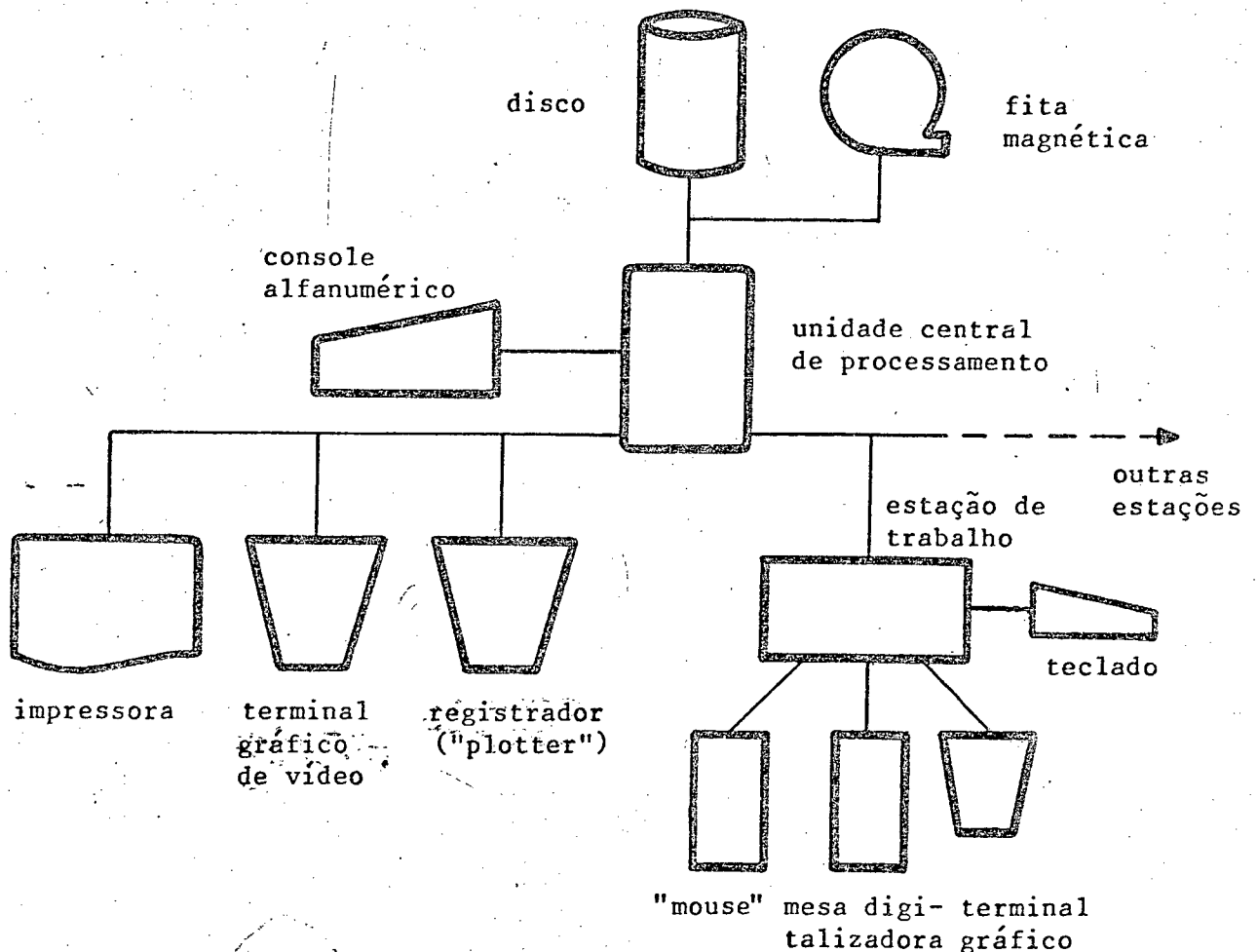


Figura 1 - Configuração hardware típica  
para um sistema PAC

Atualmente as configurações "isoladas" na forma de estações de trabalho de menor custo (baseadas em microcomputadores compatíveis IBM-PC, ou em supermicros ou minicomputadores operando com sistema operacional UNIX, tais como as estações de trabalho SUN, APOLO, etc.) estão substituindo as configurações do tipo apresentado pela figura acima.

Estas estações são dotadas dos mesmos tipos de periféricos das configurações de maior porte, possuindo no mínimo um terminal gráfico com possibilidades de várias janelas, um dispositivo de entrada do tipo "mouse", um disco "winchester" para armazenamento de massa e uma impressora gráfica.

Em muitos casos, as estações de trabalho são interligadas através de uma rede, o que permite o compartilhamento de alguns destes periféricos (impressora, fita magnética, disco de maior capacidade, etc), conforme mostrado na figura 2.

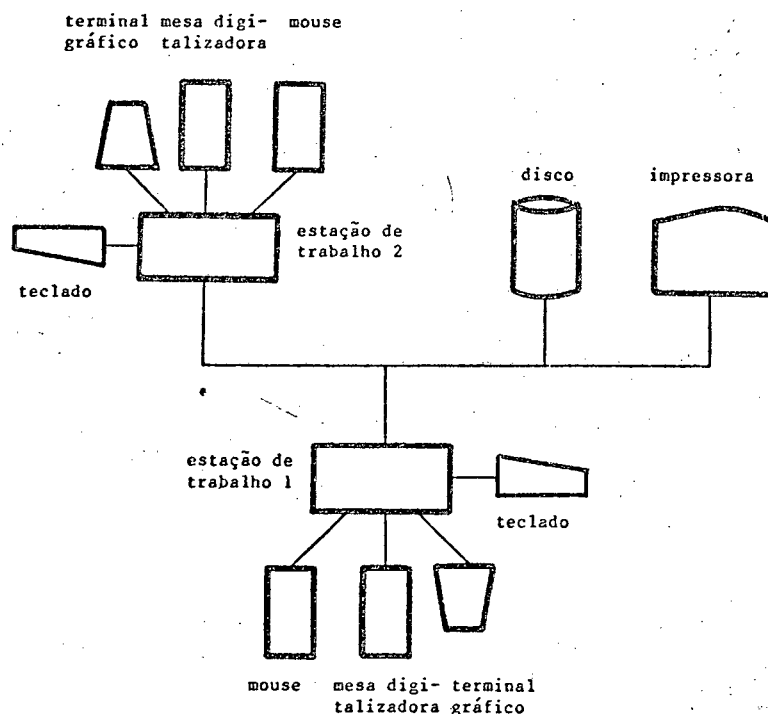


figura 2 - rede de estações de trabalho

Em termos de suporte de programação, os sistemas de PAC deverão dispor de :

- software básico: sistema operacional, compiladores, editores, depuradores, etc;
- software de apoio: pacotes gráficos, banco de dados, etc;
- software de aplicação com programas adaptados ao tipo de projeto, simuladores, etc.

## 2.3 - Projeto Assistido por Computador para Sistemas de Controle (PACSC)

### 2.3.1 - O projeto em sistemas de controle: alguns princípios

Como em outros campos da engenharia, o projeto de sistemas de controle visa a procura da alternativa de solução mais viável para um problema com objetivos pré-estabelecidos (requisitos de desempenho, etc.). Dessa forma, o processo de projeto em engenharia de controle passa pelas seguintes etapas:

#### a - Modelagem:

Nesta etapa são escolhidas as formas de descrição do comportamento do sistema. Os modelos se classificam em dois tipos:

- 1- o modelo formal, baseado numa representação matemática das características dinâmicas do sistema; este modelo pode ser construído de duas formas: a partir do comportamento observado do sistema em resposta a excitações para identificação dos parâmetros de um modelo empírico, ou a partir de um conhecimento das leis que regem o comportamento esperado do sistema;

2- o modelo conceitual, que contém informações não formuláveis, tais como limites de aplicação do modelo formal, a exatidão da representação pelo modelo do sistema real, conhecimentos intuitivos dos modos de operação.

**b - Formulação dos objetivos a atingir:**

Nesta etapa são especificados os objetivos do projeto em função de requisitos de desempenho e confiabilidade fixados pelo usuário. Algumas destas especificações são formalizáveis na forma de critérios a serem verificados.

**c - Projeto:**

Uma vez validados modelos e especificações, passa-se à etapa de projeto que se subdivide:

- 1- na análise, que consiste na realização de experimentos que permitam determinar o comportamento do sistema, seu desempenho e suas propriedades; a simulação é uma forma de verificação deste comportamento;
- 2- no projeto, propriamente dito, onde se propõe a modificação do modelo do sistema pela introdução de controladores ou leis de controle para poder atingir o comportamento pré-especificado.

As técnicas de projeto mais comuns são do tipo tentativa e erro, ou otimização automática de parâmetros do modelo (QUENC'HDU e BUISSON, 1985). Nesta etapa, a partir da criação de estratégias alternativas e da verificação do comportamento e dos efeitos destas, escolhe-se a estratégia mais adequada para os

objetivos fixados. Para tal utiliza-se, além do modelo formal, as informações conceituais armazenadas na mente do projetista e que podem hoje se encontrar embutidas no computador na forma de uma base de conhecimento, no caso de sistemas de PACSC peritos (DENHAM, 1984).

No final do projeto são fornecidas para a etapa de implementação, um conjunto de especificações da estratégia escolhida e validada. A figura 3 representa as maiores etapas do processo de projeto em sistemas de controle.

### 2.3.2 - Evolução da engenharia de controle

Os métodos de análise e projeto a disposição do engenheiro de controle tem evoluído nos últimos trinta anos desde o uso de métodos simples, às vezes empíricos, limitados a sistemas monovariáveis caracterizados por funções de transferência, até métodos mais sofisticados aplicáveis na modelagem, análise e projeto de sistemas complexos, com rígidos requisitos de desempenho e confiabilidade (ASTROM, 1983; BUNZ e GUTCHOW, 1985).

Nos primórdios da engenharia de controle, as ferramentas em uso na época (papel, lápis, régua de cálculo, computador analógico) condicionavam o tipo de técnicas utilizáveis:

- técnicas freqüenciais (Bode, Nichols, Nyquist) e lugar das raízes, que apresentavam grande capacidade gráfica;
- técnicas de cálculo aproximado evitando os cálculos extensivos, como por exemplo, o uso de correlações com a resposta em freqüência e lugar das raízes para a obtenção da resposta no tempo (ROSEMBROCK, 1974).

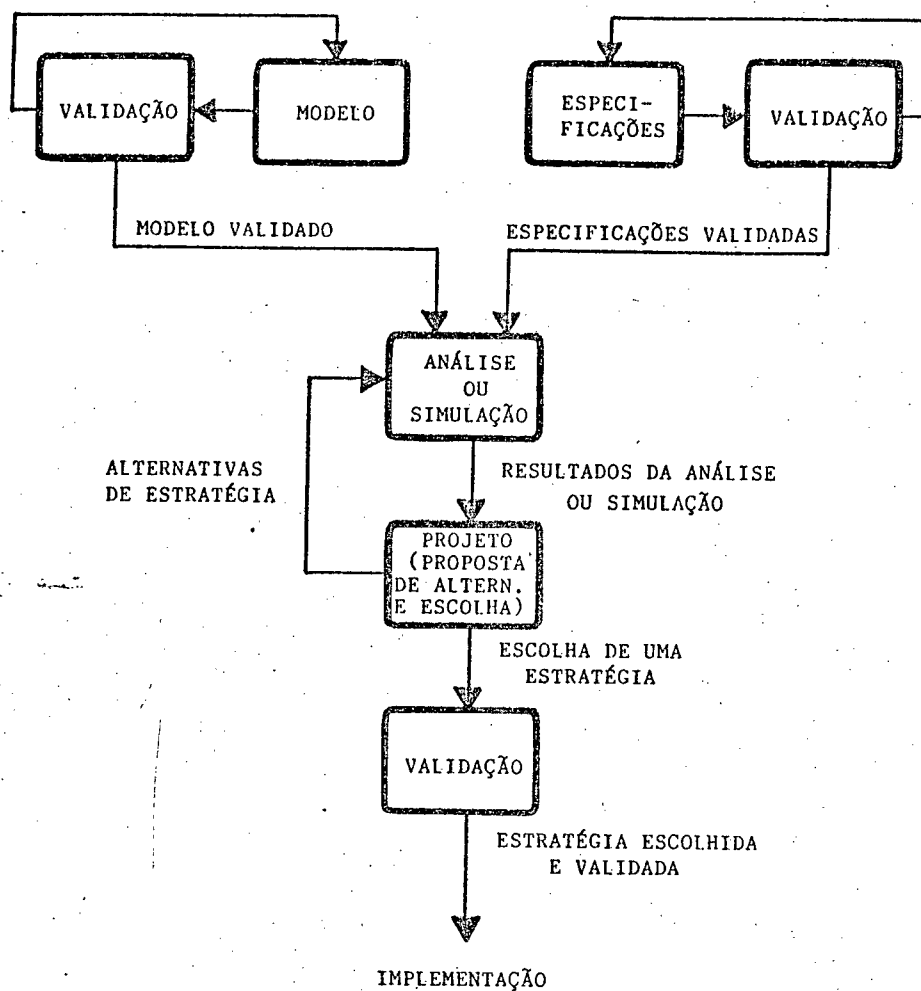


Figura 3 - Etapas do processo de projeto  
em sistemas de controle

O advento dos computadores digitais, e posteriormente a sua grande difusão, deslocaram o interesse do engenheiro de controle para o uso de métodos até então de pouco interesse, por exigir cálculos complexos e repetitivos. A partir disto, passou-se a utilizar sobretudo métodos baseados na teoria da álgebra linear, os quais permitiam estudar sistemas com maior grau de complexidade. Estes métodos passaram a ser denominados na linguagem comum do engenheiro de controle de métodos modernos, enquanto que os métodos freqüenciais e de lugar das raízes



passaram a ser denominados de métodos clássicos.

O papel do computador no processo de projeto de sistemas de controle passou a ter uma grande importância como ferramenta de apoio e de cálculo para o engenheiro de controle. Entretanto, a necessidade de se ter disponível simultaneamente vários métodos de análise e projeto (clássicos e modernos) num mesmo programa, e de aproveitar a capacidade gráfica dos computadores atuais, levou à produção de pacotes de Projeto Assistido por Computador para Sistemas de Controle (PACSC), os quais estão emergindo como ferramentas indispensáveis para o engenheiro de controle, tanto no trabalho de projeto como na sua formação.

O barateamento do custo dos computadores torna hoje viável uma grande difusão dos sistemas de PACSC no meio industrial e nos centros de pesquisa e formação e, conseqüentemente, de uma nova concepção de projeto de sistemas de controle na qual se alia a criatividade do engenheiro à potência de cálculo e à capacidade de desenhar do computador, num processo de interação constante entre o projetista e a máquina.

### 2.3.3 - Descrição do estado da arte em PACSC

Os sistemas de PACSC sofreram evoluções desde o seu aparecimento e podem ser classificados a partir das suas características comuns, em gerações de programas (GOMIDE e SZAJNER, 1985).

#### 2.3.3.1 - Primeira geração de programas

a - Características gerais:

A primeira geração de programas corresponde aos programas produzidos no período inicial de utilização do computador em sistemas de controle, e se caracterizam pelo desenvolvimento de programas orientados para implementação de métodos específicos, que tinham a função de apoiar a análise e o projeto de controle. Estes programas são utilizados de forma totalmente independente uns dos outros não possibilitando, portanto, que resultados de um programa possam ser utilizados automaticamente por outros.

b - Suporte de hardware e software:

A implementação desses programas ocorreu em grande parte, na década de 60 e início da década de 70, em computadores de grande porte ("mainframes"), limitando a sua utilização às poucas pessoas que tinham acesso a estes. A interação do usuário com o computador se dava na maioria dos casos através de cartões, fitas perfuradas ou teletipos, e em alguns poucos casos através de terminais alfanuméricos. Os resultados eram normalmente apresentados na forma de gráficos de pequena resolução em impressoras ou em registradores "off-line", após armazenamento intermediário. Os sistemas operacionais existentes para estes computadores só permitiam o processamento em batelada ("batch mode"), o que além de sub-utilizar a capacidade do computador, em particular, em termos de entrada e saída, tornava a utilização deste lenta, pois apenas um usuário podia utilizá-lo de cada vez.

2.3.3.2 - Segunda geração de programas

a - Características gerais:

A segunda geração de programas baseia-se na integração de

vários programas em pacotes e na possibilidade de interação do projetista com estes, colocando a sua disposição, de uma forma simples, um conjunto de métodos de projeto de forma a conseguir os melhores resultados. Estes pacotes surgiram em meados da década de 70, a partir do desenvolvimento ocorrido na área de computação em particular no que se refere a processadores, dispositivos gráficos de entrada e saída e sistemas operacionais.

b- Suporte de hardware e software:

Inicialmente, estes pacotes foram implementados em computadores de grande porte ("mainframes") e minicomputadores com vários terminais conectados e sistemas operacionais que permitiam o compartilhamento do computador ("time sharing"). O aparecimento dos microcomputadores propiciou um meio alternativo para desenvolvimento e implementação destes pacotes, possibilitando a criação de estações de trabalho autônomas e mais baratas; estas estações de trabalho são geralmente baseadas em microprocessadores de 16 ou 32 bits, com sistemas operacionais de fácil operacionalidade e terminais gráficos de alta resolução.

Se por um lado as estações de trabalho baseadas em microcomputadores não tem a capacidade de memória e de processamento daquelas baseadas em computadores de maior porte, o seu custo bem inferior aliado às suas potencialidades tornam-nas de grande utilidade e de grande difusão, incentivando o estudo e o desenvolvimento de pacotes de PACSC para este tipo de estação de trabalho.

A importância cada vez maior no sentido de facilitar a interface entre o projetista e o computador oferecendo a este

dispositivos de entrada/saída gráficos, levou a dotar estas estações de monitores de vídeo gráficos de alta resolução, impressoras gráficas, registradores, "light pen", mesas digitalizadoras, etc.

c - Descrição geral dos pacotes:

De acordo com a sua origem, pode-se distinguir dois tipos de pacotes:

- os pacotes obtidos a partir da integração de programas da primeira geração, através da criação de uma base de dados comum, de uma interface entre os programas e de uma interface homem-máquina orientada para o usuário e adaptada aos dispositivos de entrada/saída disponíveis; estes pacotes constituem uma evolução natural dos programas da primeira geração;
- os pacotes obtidos a partir de um projeto global do pacote de PACSC, implicando no desenvolvimento simultâneo das bibliotecas de métodos, da base de dados, da interface homem-máquina e do gerenciador que supervisionará todo o pacote; este tipo de pacote apresenta uma série de vantagens em relação ao anterior, tais como maior estruturação, uso mais racional das potencialidades (memória, processamento) da estação de trabalho, e maior facilidades de expansão.

Entre os pacotes do primeiro tipo encontram-se aqueles dedicados a uma tarefa específica do projeto tais como, por exemplo, IDPAC e SIMNON (ASTRON, 1983). IDPAC está dirigido à identificação, oferecendo como possibilidades a análise de

dados, a análise espectral, correlações, a estimação de parâmetros e a validação de modelos; SIMNON destina-se a simulação e otimização de sistemas não-lineares contínuos ou discretos, com possibilidade de escolha do método de simulação. O projeto nestes casos necessita do uso de vários destes pacotes, às vezes de forma coordenada por um programa supervisor como é o caso do "General Electric Federated System" (SPANG, 1984); este sistema é composto de 4 pacotes para identificação, simulação não linear, projeto no domínio do tempo e projeto no domínio da frequência, respectivamente.

Os pacotes do segundo tipo diferem entre si principalmente pelo conjunto de métodos (clássicos e modernos) que eles colocam a disposição do projetista para análise, simulação e projeto; alguns destes pacotes oferecem, além destes métodos, outras facilidades, tais como: métodos de análise de dados e identificação, otimização de controladores, ferramentas de teste e geração de código em tempo real. As possibilidades oferecidas por dois pacotes deste tipo, respectivamente, KEDDC (SCHMID, 1985) e SIRENA (QUENC'H DU e BUISSON, 1985), são apresentadas a seguir.

- KEDDC

- métodos para análise e síntese de sistemas SISO e MIMO lineares e não lineares, contínuos e discretos, representados no espaço de estado ou no domínio da frequência;
- métodos de identificação determinísticos e estocásticos;
- métodos para projeto de controlador: procedimentos

- clássicos, técnicas IQ, posicionamento de polos, observadores, filtro de Kalman, Nyquist inverso;
- métodos para controle adaptativo: procedimentos clássicos, modelo de referência, reguladores auto-sintonizantes;
- métodos utilitários: lugar das raízes, simulação, transformação de modelos.

- SIRENA

- métodos para análise, projeto e simulação de sistemas contínuos, discretos, amostrados, lineares e não lineares;
- cálculo formal;
- resposta freqüencial;
- traçado do lugar das raízes;
- projeto de filtros digitais;
- métodos para processamento de sinais;
- métodos para identificação.

As características de outros pacotes existentes podem ser encontradas na listagem apresentada em (JAMSHIDI e HERGET, 1985).

d - Interface homem-máquina:

Uma maior difusão dos pacotes de PACSC no meio acadêmico e industrial e o uso adequado de todas as suas potencialidades, depende fundamentalmente da forma como é realizada a comunicação entre o usuário e o pacote. Esta preocupação está bem presente no desenvolvimento dos pacotes de PACSC da segunda geração, nos quais é dada uma importância crescente à elaboração de uma interface homem-máquina que alie um alto grau de funcionalidade

com a simplicidade de uso.

A interface homem-máquina pode assumir várias formas: diálogo, linguagem de comandos, gráfica. A forma mais simples baseia-se num diálogo do tipo pergunta-resposta acrescido em alguns casos de comandos selecionados por menus (QUENC'HOU e BUISSON, 1985). Este tipo de interface mixta (menu + pergunta-resposta) é melhorada pelo uso de janelas que podem se superpor e se mover na tela do monitor de vídeo, cada uma delas sendo associadas a uma tarefa (gráficos, diálogo de entrada, métodos diversos de projeto, etc); quando o ambiente (sistema operacional) o permite, cada uma das tarefas correspondendo às janelas podem ser executadas concorrentemente (SCHMID, 1985).

Entretanto, o uso do diálogo do tipo pergunta-resposta é muito frustrante para um usuário experimentado e tem um efeito negativo no sentido da difusão de tais pacotes. A preocupação no sentido de devolver a iniciativa ao usuário levou a outras formas de comunicação, baseadas no uso de linguagens orientadas para o problema (ASTRON, 1983 ; LITTLE et al., 1985). Estas linguagens tem uma sintaxe ligada à abordagem de controle, facilitando a comunicação com o projetista experiente pela apresentação das operações a serem realizadas na forma usual dos métodos de projeto em controle, e pela construção de seqüências de comandos para formar macro comandos; para facilitar o uso por usuários inexperientes, a linguagem, além de um nível de apoio, deve conter um conjunto de comandos de fácil aprendizagem que permitam realizar as operações básicas de um projeto. Outrossim, as interfaces destes pacotes podem apresentar opções suplementares tais como: opções "default" para usuários ocasionais ou

inexperientes, o uso de editores de linha para facilitar o diálogo e a introdução de comandos e o uso de representações gráficas para introdução de modelos (diagramas de blocos, diagramas de fluxo) (PINE et al., 1985).

e - Limitações:

Os pacotes de PACSC da segunda geração apresentam limitações principalmente no que concerne à sua utilização por usuários que possuam uma experiência restrita em projeto de controle e em pacotes de PACSC, por causa da (GOMIDE e SZAJNER, 1985):

- falta de uma boa orientação para a formulação adequada do problema e obtenção das soluções mais satisfatórias;
- falta de uma boa documentação a respeito do processo de projeto utilizado e das suas limitações;
- falta de apoio na validação e implementação do projeto final.

Outras limitações encontradas dizem respeito à especialização em algum aspecto do projeto por alguns pacotes atualmente existentes e a impossibilidade de levar em conta a visão não formalizável do processo de projeto.

#### 2.3.3.3 - Terceira geração de programas

Algumas das limitações anteriormente citadas para os pacotes de PACSC da segunda geração poderão ser superadas através dos pacotes da terceira geração que, além de oferecer ampla variedade de métodos de projeto, permitirão a orientação e o apoio do usuário mesmo que não especialista. Estes pacotes baseiam-se em conceitos de inteligência artificial e permitem o uso de



informações conceituais importantes no projeto, as quais não podem ser representadas matematicamente, e que facilitarão as operações nas quais a intuição, o julgamento e a capacidade do projetista são fundamentais. Desta forma será oferecido ao projetista um sistema de apoio especialista cujo o papel não é substituí-lo mas orientá-lo nas suas escolhas (métodos, estratégias), nas suas interpretações de resultados e na sua procura da solução ótima, a partir do conhecimento da experiência acumulada de especialistas e do uso de informações não formalizáveis (DENHAN, 1984); estes tipos de pacotes permitirão uma melhoria do projeto, uma redução da probabilidade de erros, uma melhor organização e documentação do processo de projeto, a possibilidade de validação dos resultados do projeto, e uma aprendizagem mais rápida do processo de projeto pelo usuário (TAYLOR e FREDERICK, 1984).

Este tipo de pacote se encontra ainda na sua fase inicial, existindo já alguns exemplos simples destes: sistema perito de projeto de compensadores avanço-atraso (JAMES et al., 1985) é um exemplo das possibilidades oferecidas e do futuro promissor deste tipo de pacote.

Na figura 4 é apresentado um diagrama esquemático de uma estrutura para um pacote da terceira geração proposto por Taylor e Frederick (TAYLOR e FREDERICK, 1984).

#### 2.4 - Os programas de PACSC no ensino da engenharia de controle

O uso de pacotes de PACSC é importante não somente a nível do seu uso industrial mas também na formação do engenheiro de controle. Estes pacotes, principalmente na sua parte de

comunicação homem-máquina, podem vir a ser modificados em função deste último objetivo.

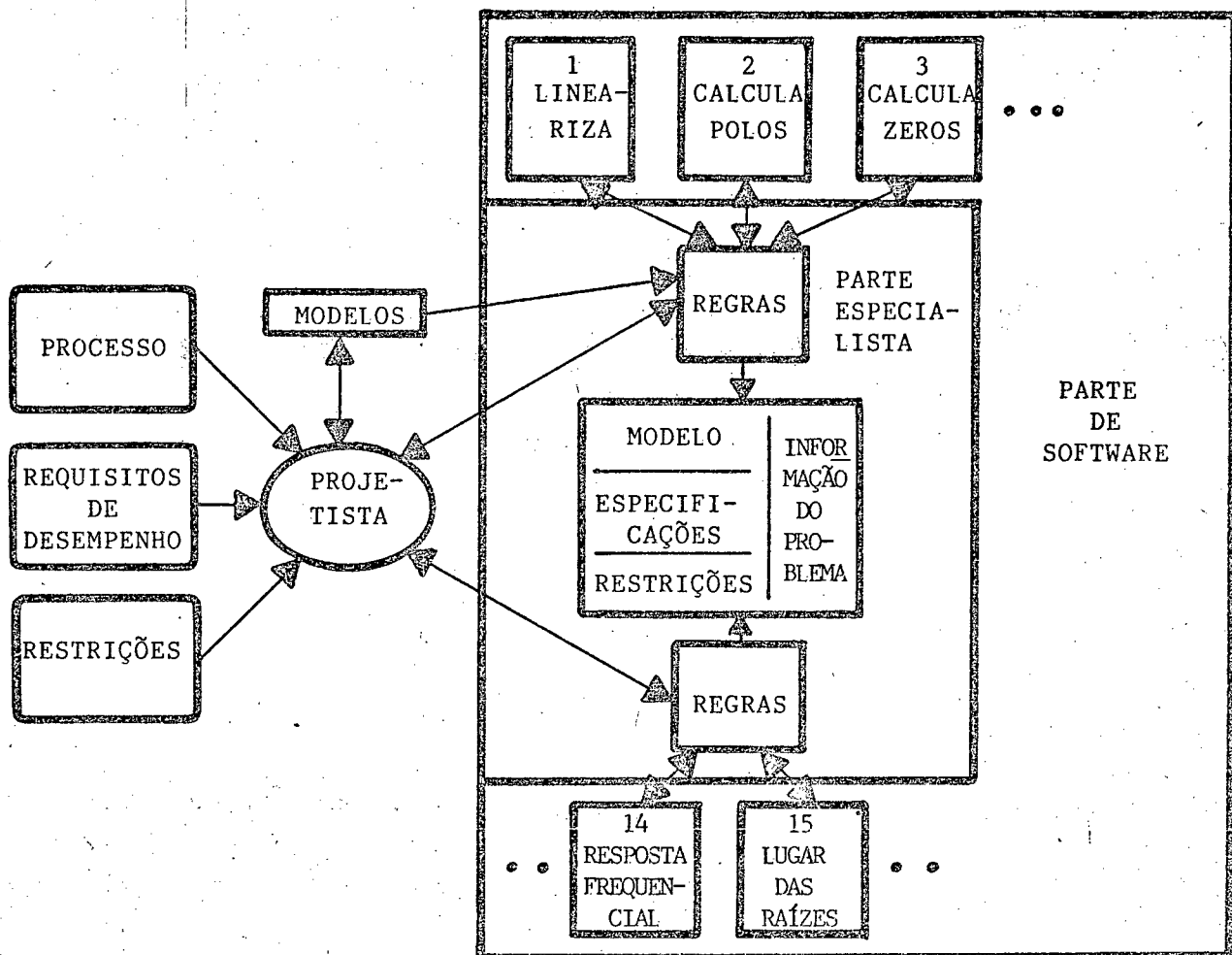


figura 4 - estrutura para um pacote da terceira geração

As vantagens trazidas pelo seu uso são numerosas, entre as quais pode-se citar:

- a possibilidade de comparação de diferentes métodos;
- a possibilidade de verificação da capacidade e das limitações das técnicas existentes;

- a economia de tempo, evitando a necessidade de realização de cálculos extensos e a realização de gráficos;
- a apresentação dos mais recentes desenvolvimentos da teoria de controle;
- a aquisição de habilidade e experiência na realização do projeto de controle.

## 2.5 - A evolução dos programas de PACSC no LCMI

### 2.5.1 - Programas desenvolvidos no LCMI

O início dos trabalhos do Laboratório de Controle e Microinformática (LCMI) da UFSC na área de programas de PACSC, ocorreu em 1983 com o desenvolvimento de um primeiro programa que visava a análise, o projeto e a simulação de sistemas lineares monovariáveis contínuos utilizando a teoria de controle clássica (ZENI et al., 1984); ele foi implementado em um microcomputador de 8 bits, tendo o seu uso dirigido para ensino e possuindo como principais limitações:

- a sua pequena resolução gráfica;
- o uso de uma comunicação do tipo pergunta-resposta;
- a baixa velocidade de processamento.

Em meados de 1985 uma nova versão deste programa começou a ser desenvolvida para implementação em um microcomputador de 16 bits compatível IBM-PC. Esta versão, concluída em fins de 1986, propiciou uma melhora significativa em todos os aspectos do programa, destacando-se (FARINES et al., 1986):

- o aumento da velocidade de processamento;
- a melhora da comunicação com o usuário pelo uso conjugado de menus e diálogos;

- a extensão da utilização das ferramentas da teoria clássica para o caso de sistemas discretos;
- a melhoria dos algoritmos utilizados;
- as possibilidades oferecidas de visualização da transformação de modelos;
- o melhor aproveitamento das capacidades gráficas do computador.

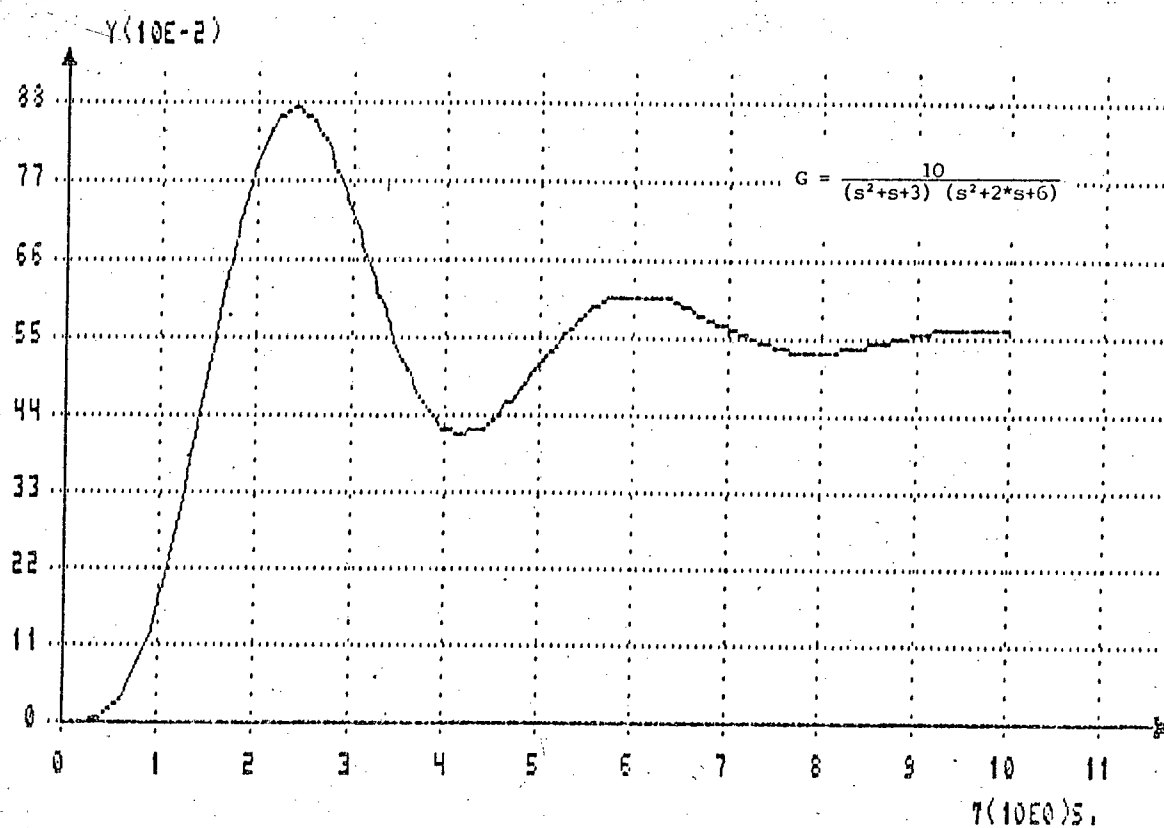
Esta versão se encontra hoje em pleno uso, devendo ser utilizada principalmente como uma ferramenta de apoio aos cursos de graduação e de pós-graduação da UFSC e de outras universidades para as quais ela está sendo fornecida.

A figura 5 apresenta curvas resultantes da utilização das diversas ferramentas disponíveis no programa.

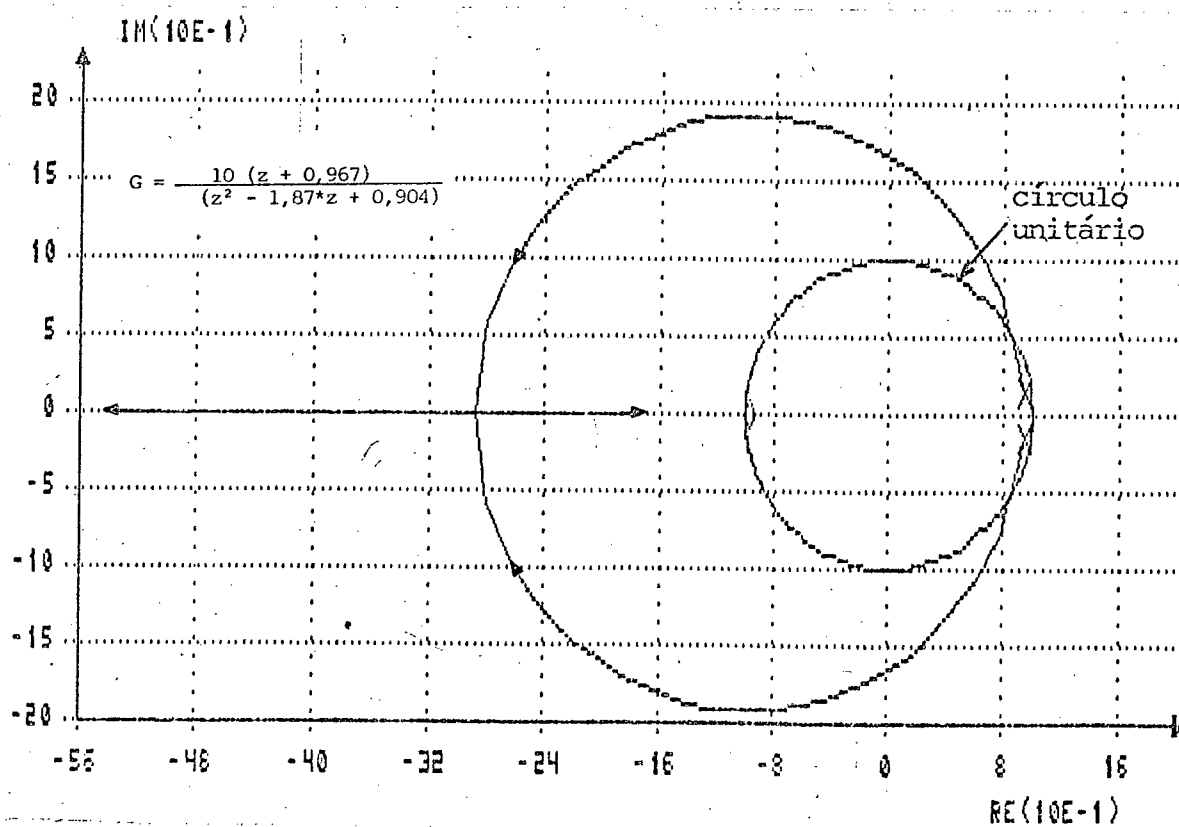
#### 2.5.2 - O programa proposto

A tendência natural de evolução no desenvolvimento destes programas levou ao projeto de uma nova versão com objetivos bem mais abrangentes do que as versões anteriores, a qual é objeto deste trabalho.

Esta versão pretende superar as limitações das versões anteriores, através do fornecimento de um conjunto mais amplo de métodos (clássicos e modernos) para análise e projeto, e da criação de uma interface homem-máquina mais eficiente e otimizada. Ela será mais abrangente também em termos da representação e análise de sistemas com estruturas mais complexas, sendo estendida para o caso de sistemas multivariáveis.

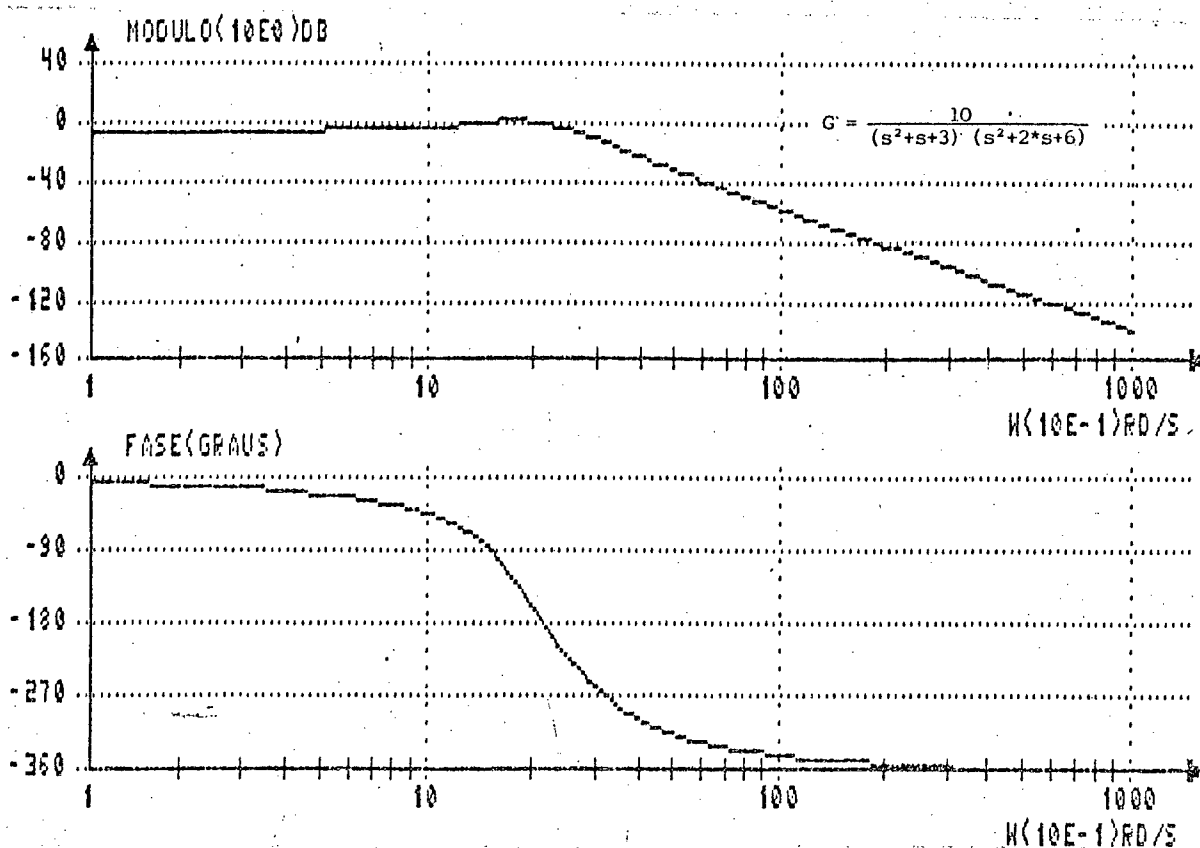


a) simulação

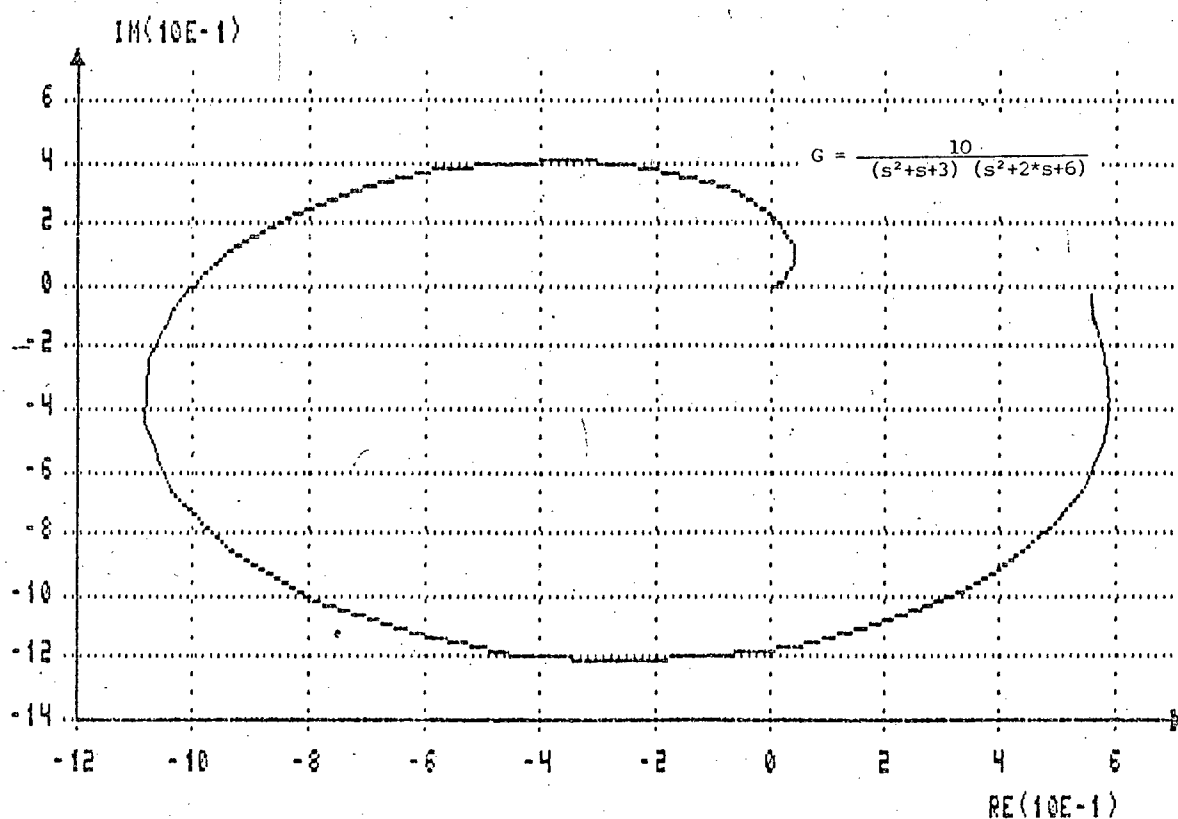


b) lugar das raízes discreto

figura 5 : curvas resultantes da utilização de ferramentas do programa



c) diagramas de Bode



d) diagrama de Nyquist

figura 5 (cont.) : curvas resultantes da utilização  
de ferramentas do programa

Dentre as características desta nova versão, podem também ser citadas:

- a possibilidade de transformação de modelos e redução formal de diagramas de blocos;
- a possibilidade da introdução de blocos não lineares e atraso;
- a utilização de um editor de textos para facilitar a entrada de dados;
- as várias possibilidades de traçado e manipulação dos gráficos de saída.

Da mesma forma que na versão anterior, o desenvolvimento será realizado em microcomputador de 16 bits compatível IBM-PC.

## 2.6 - Conclusão

Nesta primeira parte foi feita uma rápida apresentação do estado da arte na área de PACSC e discutida a importância destas novas técnicas para a análise e o projeto em engenharia e controle. É também apresentada a proposta deste trabalho dentro da perspectiva da continuação de trabalhos anteriormente realizados no ICMI.

Na segunda parte deste trabalho, será apresentada a especificação detalhada e elementos do projeto do programa de PACSC proposto.

## P A R T E 2

UM PROGRAMA DE PROJETO ASSISTIDO POR COMPUTADOR  
PARA SISTEMAS DE CONTROLE



## C A P Í T U L O 3

### METODOLOGIA ADOTADA

#### 3.1 - Introdução

O objetivo desejado é o desenvolvimento de um programa interativo de PACSC para análise e projeto de sistemas mono e multivariáveis, contínuos e discretos.

A realização de um programa deste tipo envolve a solução de um grande número de problemas, que vão desde os relacionados com as tarefas de apóio à implementação (gráficos, armazenamento de dados, mecanismos da interface homem-máquina, etc.) aos relacionados com a implementação do conjunto de métodos de projeto propriamente dito (estudo e implementação dos algoritmos, análise da estabilidade numérica destes, etc.).

Na avaliação do esforço a ser dispendido para desenvolver o programa proposto, deve-se levar em conta, além da complexidade e da diversidade de tarefas, alguns dados colhidos para programas semelhantes a este. Por exemplo, Astrom (ASTROM, 1983) apresenta os seguintes dados mostrados na tabela 1; as características dos programas citados nesta tabela podem ser encontradas em (JAMSHIDI e HERGET, 1985).

Levantamentos feitos no próprio ICMI a respeito do

desenvolvimento da versão monovariável contínua e discreta, para microcomputadores compatíveis IBM-PC, mostram que:

- o esforço foi de aproximadamente 6 homens/mês;
- grande parte do esforço dispendido se encontra na fase de testes e validação do programa.

Falhas no desenvolvimento do programa, como a não-utilização de uma metodologia que possibilitasse em uma maior estruturação do programa e a insuficiência da documentação, são causas de um maior esforço.

NOME	NÚMERO DE LINHAS DE CÓDIGO	TAMANHO DO PROGRAMA Kbytes
IDPAC	37.000	470
MODPAC	41.000	570
SIMNON	25.000	360
POLPAC	32.000	460

Tabela 1 : exemplos do tamanho de programas.

Baseado nestas constatações e em estudos mais gerais (PRESSMAN, 1982), concluiu-se que era necessário para o desenvolvimento deste programa:

- usar uma metodologia de desenvolvimento do software;
- formar uma equipe de trabalho que atuasse de forma coordenada.

### 3.2 - Metodologia de desenvolvimento adotada

Neste trabalho, utilizou-se a mesma metodologia de desenvolvimento que vem sendo adotada no LCMI, a qual é descrita em (FARINES, 1986-2) e cujos detalhes podem ser encontrados nos clássicos da literatura de Engenharia de Software (PRESSMAN, 1982; ROSS, 1977; YOURDON e CONSTANTINE, 1978). A seguir,

apresenta-se algumas informações a seu respeito que descrevem-na sucintamente.

### 3.2.1 - Detalhes sobre a metodologia

O ciclo de vida do software se divide em três fases principais: planejamento, desenvolvimento e manutenção (PRESSMAN, 1982). Cada uma dessas fases se subdivide em etapas, para as quais um conjunto de métodos e técnicas são disponíveis; ao final de cada etapa tem-se como produto um documento contendo os resultados desta.

O planejamento inicia a partir de um pedido do usuário, estando dividido nas seguintes etapas:

- definição do sistema: nesta etapa determina-se as alternativas de solução para o problema proposto, estuda-se a realizabilidade destas soluções e escolhe-se a alternativa mais adequada;
- planejamento do software: visa estimar os recursos, os custos e o cronograma de desenvolvimento do software correspondente à implementação da solução escolhida;
- análise dos requisitos: esta etapa visa a identificação dos requisitos funcionais e operacionais da solução escolhida, o levantamento das especificações funcionais do software e a sua validação, bem como o estabelecimento dos testes de validação do programa. A técnica de análise dos requisitos e especificação do software adotada neste trabalho é a técnica "Structured Analysis and Design Techniques - S.A.D.T." apresentada em (ROSS, 1977);

O desenvolvimento do software é realizado a partir dos requisitos do software e das especificações funcionais formais obtidas na fase anterior, sendo composto das seguintes etapas:

- projeto preliminar: nesta etapa estabelece-se a relação entre os vários elementos da solução do software (módulos), apresenta-se toda a estrutura de controle destes e suas interfaces, e prepara-se também a etapa dos testes de integração;
- projeto detalhado: nesta etapa cada módulo é explicitado e detalhado em termos de procedimentos e prepara-se a etapa de testes de unidade;
- codificação: nesta etapa, a partir do projeto detalhado, cada módulo é codificado na linguagem escolhida;
- implementação e testes:
  - a - testes de unidade: testa-se cada módulo individualmente;
  - b - testes de integração: integra-se os módulos verificando-se principalmente o interfaceamento entre eles;
  - c - testes de validação: verifica-se se o software produzido está em conformidade com os requisitos expressos na fase de planejamento;
  - d - teste do sistema: o software aprovado é integrado com os outros elementos ( hardware ), realizando-se testes de integração e aprovação;
- instalação do sistema: nesta etapa o sistema aprovado é instalado "in-loco" e testes exaustivos de aceitação são realizados.

A fase de manutenção inicia após a instalação, podendo existir quatro tipos diferentes:

- manutenção corretiva: para corrigir erros detetados durante a operação do sistema;
- manutenção adaptativa: para adaptar o sistema a novas condições (novo hardware, novos periféricos, etc);
- manutenção de aperfeiçoamento: para aperfeiçoar as funções existentes ou introduzir novas funções;
- manutenção preventiva: para melhorar o software com o objetivo de evitar futuras manutenções ou de torná-lo mais confiável.

### 3.2.2 - Sistemática de trabalho

O trabalho apresentado nesta dissertação se coloca como uma das ações (ação 2) de um trabalho mais amplo em desenvolvimento no LCMI, denominado de operação CAD. Esta operação está dividida em seis ações:

- ação 1: manutenção e aperfeiçoamento do pacote de PACSC para sistemas monovariáveis já desenvolvido;
- ação 2: desenvolvimento de um pacote de PACSC para sistemas mono e multivariáveis, utilizando uma interface baseada em menus e diálogos;
- ação 3: desenvolvimento de um pacote de PACSC para sistemas mono e multivariáveis, utilizando uma interface baseada em uma linguagem de comandos;
- ação 4: desenvolvimento de um pacote de identificação assistida por computador;
- ação 5: estudo e implementação de algoritmos para os métodos de controle moderno;

- ação 6: estudo e implementação de rotinas para o traçado e manipulação gráficas.

A realização desta operação envolve uma equipe de trabalho, onde cada membro participa em uma das ações citadas acima. Um dos membros tem o papel de supervisor da equipe, o qual é encarregado do planejamento, coordenação e supervisão do trabalho dos outros membros da equipe em função de um cronograma previamente estabelecido. Reuniões semanais são realizadas com o objetivo de informar e debater as atividades realizadas em cada uma das ações, apresentar as dificuldades encontradas e planejar o trabalho da semana seguinte.

### 3.3 - Conclusão

Neste capítulo foram apresentadas algumas considerações que indicam a necessidade do uso de uma metodologia de desenvolvimento e da formação e organização de uma equipe de trabalho para a realização do programa proposto. Em seguida, uma descrição sucinta da metodologia de desenvolvimento foi apresentada com o objetivo de introduzir os conceitos na qual ela está baseada. Nos próximos capítulos serão apresentadas as especificações e alguns aspectos do projeto do programa a ser desenvolvido.

## C A P Í T U L O 4

### DEFINIÇÃO DO PROGRAMA

#### 4.1 - Introdução

Este capítulo apresenta o objetivo e as características principais do programa, dando uma descrição geral das suas partes interativas e operativas e destacando, a seguir, os mecanismos de interação entre usuário e programa e a linguagem de comunicação escolhida.

#### 4.2 - Objetivo

O programa deve integrar um conjunto de ferramentas para análise, projeto e simulação de sistemas monovariáveis e multivariáveis, contínuos e discretos; é também possível a utilização de modelos de blocos do processo armazenados em arquivo, obtidos por identificação através da utilização de um programa específico em desenvolvimento. A integração dessas ferramentas deve ser realizada de forma a adaptar-se ao modo de trabalho do projetista de controle, tanto na forma de utilização das ferramentas como na visualização dos respectivos resultados.

#### 4.3 - Descrição geral do programa

O programa se divide em duas partes:

- uma parte interativa, que permite a interação do usuário com o programa;
- uma parte operativa, que realiza todas as operações internas escolhidas pelo usuário e durante a qual este não tem ação possível.

A figura 1 representa as diversas ações do programa, as quais podem ser comparadas às ações de uma máquina qualquer:

- a interação, subdividida em:
  - introdução do objeto a tratar, no caso presente corresponde à declaração do sistema;
  - escolha da ferramenta, no caso a escolha do método de projeto e a introdução de todos os dados necessários à sua execução;
  - recepção do objeto resultante, no caso visualização dos resultados obtidos no uso do método de projeto escolhido;
- a operação, que no caso do programa equivale ao uso do método escolhido para o sistema declarado.

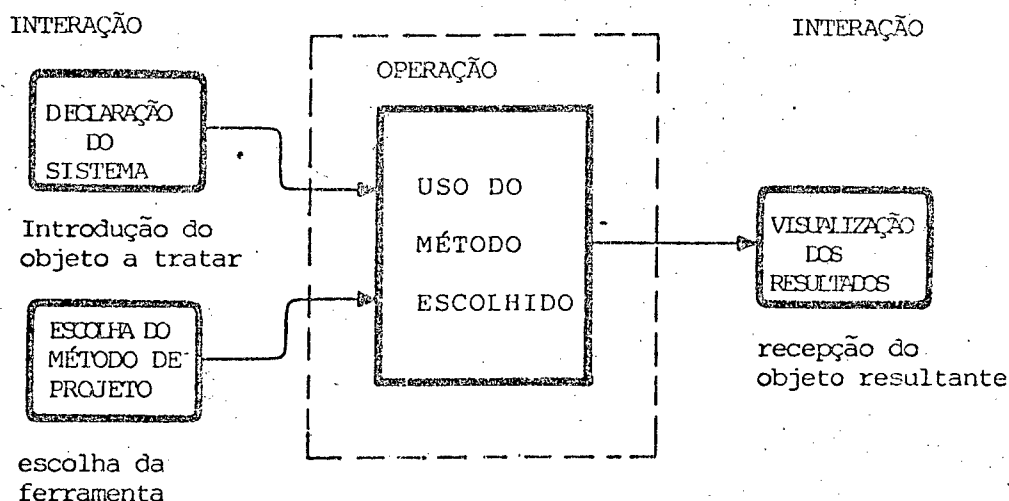


Figura 1 -- Representação das ações do programa



#### 4.4 - Descrição geral da parte interativa do programa

##### 4.4.1 - Introdução do objeto a tratar: declaração do sistema

A declaração dos dados que definem o modelo do sistema a ser estudado é realizada declarando-se, separadamente, as duas partes que o compõem conforme apresentado na figura 2:

- a parte referente ao processo;
- a parte referente ao controle.

A parte referente à medição pode ser declarada junto com o processo ou com o controle.

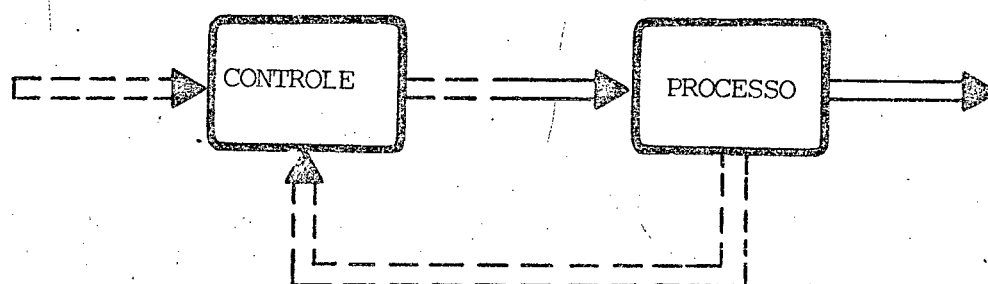


Figura 2 - Estrutura de declaração do sistema

Essa forma de declaração permite ao usuário poder utilizar as ferramentas do programa tanto na análise do processo isoladamente como na análise do sistema global (processo + controle). Para tal, os modelos do processo e do controle são armazenados internamente em estruturas de dados separadas, sendo essas estruturas interligadas automaticamente pelo programa de modo a formar a estrutura do sistema.

Os sistemas podem ser contínuos (processo e controle contínuos), discretos (processo e controle discretos) ou híbridos

(processo contínuo e controle discreto). Os modelos são declarados através da representação por funções de transferência ( $S$  ou  $Z$ ), ou da representação no domínio do tempo por equações de estado (contínuas ou discretas). As declarações de sistemas misturando-se os dois domínios de representação (tempo e frequência), serão permitidas somente no caso de sistemas monovariáveis que não possuam perturbação. O programa permite ainda ao usuário declarar simultaneamente um processo e até cinco estruturas de controle diferentes; durante a execução do programa o usuário seleciona, através do respectivo nome, o sistema para o qual a ferramenta escolhida será utilizada.

#### 4.4.2 - Escolha das ferramentas

As ferramentas disponíveis no programa se classificam em ferramentas de cálculo e ferramentas auxiliares. Após a seleção de cada ferramenta, devem ser definidos os dados necessários à utilização desta.

##### a - Ferramentas de cálculo

As ferramentas de cálculo se dividem em: ferramentas clássicas, ferramentas modernas, verificação das propriedades, simulação e transformação de modelos.

As ferramentas clássicas são utilizadas somente no caso de sistemas monovariáveis e englobam os seguintes métodos:

- resposta frequencial (para sistemas contínuos): Bode, Nichols, Nyquist;
- lugar das raízes (para sistemas contínuos e discretos).

As ferramentas modernas são utilizadas tanto no caso mono

como multivariável, com as seguintes possibilidades:

- realimentação de estados:
  - posicionamento de polos;
  - minimização de critério quadrático (Riccati);
- cálculo de observadores;
- servocompensador robusto.

A verificação das propriedades inclui as seguintes possibilidades:

- estabilidade;
- controlabilidade;
- observabilidade.

A simulação pode ser utilizada para qualquer tipo de sistema, podendo ser realizada de duas formas distintas:

- a simulação global;
- a simulação passo a passo (bloco a bloco).

A escolha da forma de realização da simulação é feita automaticamente pelo programa, em função do tipo de sistema, da presença de não-linearidades, etc.

A transformação de modelos permite, quando possível, obter representações equivalentes do modelo inicial do processo (função de transferência  $\Leftrightarrow$  equações de estado), ou discretizá-lo. É permitido também realizar a redução formal de diagramas de blocos para a obtenção da função de transferência global.

#### b - Ferramentas auxiliares

As ferramentas auxiliares do programa possibilitam o auxílio ao usuário a respeito da utilização das ferramentas disponíveis e

a informação na tela dos modelos do processo ou do controle que estão sendo utilizados.

#### 4.4.3 - Visualização dos resultados

Os resultados obtidos na utilização de uma ferramenta podem ser apresentados na forma de:

- gráficos na tela;
- gráficos no plotter;
- gráficos na impressora.

As possibilidades oferecidas em termos de gráficos são as seguintes:

- diagramas de Bode (amplitude e fase);
- diagramas de Nyquist e de Nichols;
- diagramas no tempo (X-T);
- diagramas entre variáveis (X-Y);
- diagramas do lugar das raízes (contínuo e discreto).

##### a - Gráficos na tela

Os resultados são sempre apresentados automaticamente na forma de gráficos na tela no fim da utilização de uma ferramenta. Posteriormente, estes podem ser gravados em disco para futura reapresentação. É também possível realizar operações sobre os gráficos, tais como:

- introdução de texto;
- leitura de pontos do gráfico;
- realização de ampliações ("ZOOM");
- introdução de curvas-padrões (circulo 1.3 no diagrama de Nyquist, retas de amortecimento constante no lugar

das raízes, etc.).

As modificações introduzidas podem eventualmente ser gravadas com o gráfico.

Os gráficos na tela podem se apresentar na forma de:

i) uma única janela, contendo :

- uma única curva;
- várias curvas de uma mesma variável correspondendo a vários valores de um parâmetro;

ii) várias janelas, contendo cada uma:

- curvas referentes a diversas variáveis do sistema;
- diferentes tipos de curvas (Bode, resposta no tempo, etc.) correspondendo a resultados gravados em análises anteriores.

O usuário define, dependendo do caso, o número de janelas e as variáveis ou relações a serem apresentadas em cada uma delas, assim como o tipo de traçado (contínuo, pontilhado, etc) para cada uma das curvas de cada janela.

#### b - Gráficos no "plotter"

Esta possibilidade de saída permite reapresentar, na forma de traçado no plotter, resultados previamente gravados em disco, podendo ser apresentados simultaneamente diferentes tipos de curvas.

#### c - Gráficos e tabelas na impressora

É possível também obter um relatório na impressora que

possibilita visualizar os resultados na forma de gráficos e tabelas, com a possibilidade de adicionar as seguintes informações:

- um cabeçalho escrito pelo usuário;
- os dados do processo e do controle utilizados;
- os dados utilizados no uso da ferramenta (Ex: valores dos parâmetros, do tempo de simulação, etc.).

#### 4.5 - Descrição detalhada da parte interativa do programa: Interface Homem-Máquina

A parte interativa do programa está organizada na forma de uma Interface Homem-Máquina que permite ao usuário interagir com o programa na sua inicialização, na declaração do sistema, na escolha das ferramentas e na visualização dos resultados. Os princípios de comunicação utilizados nesta interface são apresentados a seguir.

##### 4.5.1 - Janelas de comunicação

A tela de comunicação do programa se apresenta dividida em várias janelas, possibilitando com isto a apresentação simultânea de diferentes informações e aumentando a eficiência da interface homem-máquina. A figura 3 apresenta a forma da tela de interação do programa, onde cada uma das janelas tem a seguinte finalidade:

- janela 1 : janela para realização do diálogo e apresentação de resultados;
- janela 2 : janela para apresentação dos menus;
- janela 3 : janela para seleção das ferramentas (introdução e/ou visualização dos códigos das diretivas), e para apresentação de dados

lidos no gráfico;

- janela 4 : janela para apresentação de códigos de erro, e para a indicação de mensagens de leitura de tela;
- janela 5 : janela temporária superposta à janela 1, para apresentação de auxílio a respeito do uso das ferramentas ou de eventuais erros ocorridos.

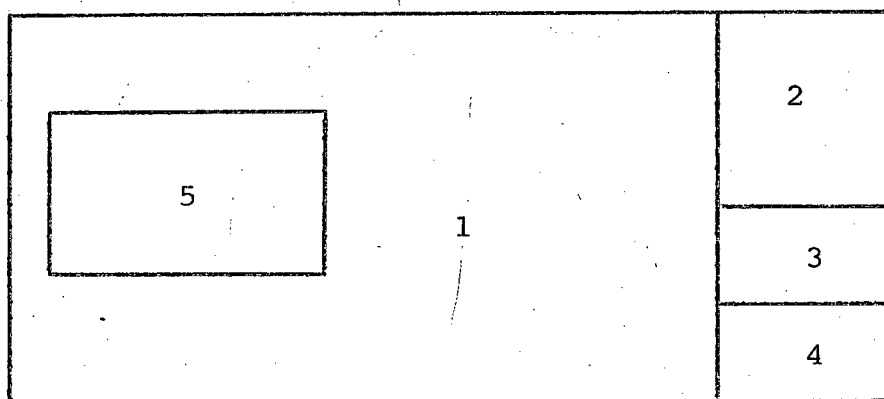


Figura 3 - Forma da tela de comunicação do programa

#### 4.5.2 - Mecanismos da comunicação homem-máquina

A forma escolhida para a comunicação entre o usuário e o programa é baseada no uso conjugado de:

- diretivas escolhidas em menus ou introduzidas na forma de um código;
- diálogos do tipo pergunta-resposta.

A verificação da consistência dos dados introduzidos é realizada pelo gerenciador do programa, descrito em outro capítulo, que, no caso de erro, fornece na janela correspondente mensagens de aviso e orientação ao usuário.

### a - Comunicação por diretivas

A seleção da ferramenta ou operação desejada pode ser realizada de duas formas diferentes:

- i) através da escolha de uma das opções do menu de seleção corrente apresentado na janela 2, as quais estão associadas às teclas de funções especiais do computador (F1,...,F10); a visualização da diretiva escolhida pode ser feita graças à escrita automática do código correspondente na janela 3;
- ii) pela introdução na janela 3 do código correspondendo a diretiva escolhida;

Os códigos utilizados são associados às opções encontradas nos menus de seleção e seguem uma terminologia adaptada aos métodos utilizados na engenharia de controle.

As diretivas do programa, separadas em grupos por função, são apresentadas nas tabelas 2, 3, 4 e 5 abaixo.

CÓDIGO	FUNÇÃO	CARACTERÍSTICAS PRINCIPAIS
DIRETIVAS DE DECLARAÇÃO		
PROC	declara modelo do processq	
CONTR	declara modelo do controle	<ul style="list-style-type: none"> <li>- necessidade de compatibilizar as variáveis de ligação do processo e controle;</li> <li>- coeficientes variáveis (até 3);</li> <li>- possibilidades de opções secundárias: ver tabela 7.</li> </ul>

Tabela 2 - diretivas de declaração



CÓDIGO	FUNÇÃO	CARACTERÍSTICAS PRINCIPAIS
DIRETIVAS DE CÁLCULO		
REDU	realiza redução formal	- fornece resultados formal e numérico.
TRANS	transforma modelo do processo	- transformações dependem do tipo do processo (contínuo ou discreto); - possibilidades de opções secundárias: ver tabela 6.
LDR	realiza lugar das raízes	- sistemas contínuos e discretos. - o coeficiente variável pode ser declarado pelo usuário ou criado pelo programa.
RFR	realiza reposta frequencial	- cálculo entre variáveis intermediárias; - sistemas contínuos.
SIMU	realiza simulação	- simula sistema completo incluindo não-linearidades.
PROP	verifica propriedades	- somente para sistemas lineares; - propriedades: estabilidade, controlabilidade e observabilidade.

Tabela 3 - diretivas de cálculo

CÓDIGO	FUNÇÃO	CARACTERÍSTICAS PRINCIPAIS
DIRETIVA DE SAÍDA		
SAÍDA	reapresenta resultados gravados	- possibilidade de diferentes tipos de curvas em cada janela, a pedido do usuário; - opções para manipulação dos gráficos disponíveis na forma de menu especial para gráfico.

Tabela 4 - diretiva de saída

CÓDIGO	FUNÇÃO	CARACTERÍSTICAS PRINCIPAIS
DIRETIVAS AUXILIARES		
INF	informa os dados do processo ou controle	- dados reapresentados na forma do diálogo original.
AUX	auxilia o usuário	- informações de apoio sobre o uso das ferramentas do programa.
FIM	termina execução do programa	- destruição dos arquivos de resultados criados no disco de trabalho.

Tabela 5 - diretivas auxiliares

As opções secundárias correspondentes às diretivas de transformação de modelos e controle são apresentadas nas tabelas 6 e 7, respectivamente.

CÓDIGO	FUNÇÃO
OPÇÕES SECUNDÁRIAS DA TRANSFORMAÇÃO	
TEC	transforma modelo do processo para equações de estado contínuas
TS	transforma modelo do processo para função de transferência contínua (S)
TED	transforma modelo do processo para equações de estado discretas
TZ	transforma modelo do processo para função de transferência discreta (Z)
TER	transforma modelo do processo para equação de recorrência

Tabela 6 - opções secundárias da transformação de modelos

CÓDIGO	FUNÇÃO
OPÇÕES SECUNDÁRIAS DO CONTROLE	
CLAS	declara controle clássico
MOD	declara controle moderno
MODI	modifica o modelo de blocos de controle
MATR	calcula matrizes (ganho de realimentações, observadores, etc)

Tabela 7 - opções secundárias do controle

## b - Comunicação por diálogo

Quando uma ferramenta for selecionada (através das diretivas), se instaura na janela 1 um diálogo do tipo pergunta-resposta, através do qual o programa requisita todos os dados necessários para o uso da ferramenta selecionada; com o objetivo de facilitar a entrada de dados, os diálogos para cada ferramenta são subdivididos em blocos, de forma a introduzir e visualizar as informações introduzidas conjuntamente. Este diálogo pode ser interrompido a qualquer instante pelo uso de uma tecla especial (escape), voltando ao menu de seleção.

## b.1 - Editor de textos

Com o objetivo de facilitar a comunicação, quando do diálogo, é colocado à disposição do usuário um editor de textos próprio do programa, adaptado ao problema.

Os comandos deste editor são os seguintes: mover o cursor (para cima, para baixo, para a direita, para a esquerda), apagar

caracteres inseridos, habilitar e desabilitar o editor; a inserção de caracteres é automática quando o editor estiver habilitado e a sua desabilitação indica o fim da resposta.

O editor do programa opera como um editor de linha para a edição ou modificação da resposta a uma determinada pergunta. Em funcionamento normal de edição, ele é automaticamente habilitado em duas situações diferentes:

- i) após cada pergunta apresentada na tela quando da realização do diálogo;
- ii) após uma pergunta qualquer, quando, após a análise dos dados pelo gerenciador, for encontrada alguma resposta inconsistente.

Ao final da resposta à última pergunta de cada bloco de diálogo, antes da análise dos dados introduzidos, é permitido ao usuário fazer modificações desses dados pelo posicionamento do cursor sobre a resposta a ser modificada e a habilitação manual do editor; a análise dos dados é autorizada pelo usuário através de dois comandos consecutivos de desabilitação do editor.

## b.2 - Diálogo de inicialização

A inicialização corresponde a uma série de operações sequenciais que permitem, através de um diálogo do tipo pergunta-resposta, atualizar informações gerais do programa. Essas operações são as seguintes:

- apresentação de documentação: apresenta, a pedido do usuário, uma documentação sucinta descrevendo os objetivos e as características do programa;
- definição da configuração: permite definir a localização

do disco de trabalho e informar as opções de saída existentes;

- dados sobre a estatística do programa: permite introduzir o código de identificação do usuário, o qual será utilizado na criação de subdiretório no disco de trabalho, e atualizar o relógio de tempo real do computador;
- declaração do processo: permite declarar o processo a ser analisado; um novo pedido de declaração, no decorrer da utilização do programa, permite substituir o processo declarado por outro.

### b.3 - Exemplos

Os dois exemplos apresentados a seguir ilustram a forma do diálogo para duas diretivas do programa: a diretiva de declaração do processo e a diretiva lugar das raízes. Nestes diálogos as partes em **negrito** correspondem aos dados introduzidos pelo usuário.

#### 1 - Declaração do processo

Seja o processo representado pelo diagrama de blocos da figura 4 abaixo:

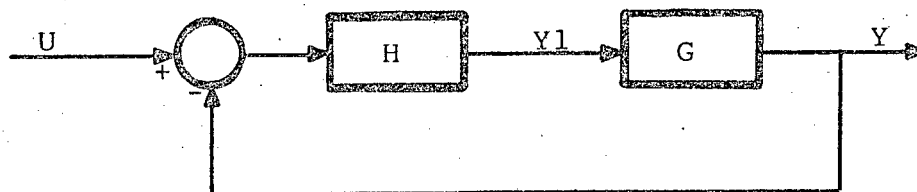


Figura 4 - Diagrama de blocos do processo

A declaração deste processo é realizada através dos blocos de diálogo apresentados na figura 5 abaixo:

primeiro bloco de diálogo:

PROCESSO Antigo Novo : N NOME DO PROCESSO : motor TIPO (Lapl; Transfz; Eec; Eed) : L ENTRADAS : u PERTURBAÇÕES : SAÍDAS : y BLOCOS LINEARES : g, h BLOCOS NÃO LINEARES : BLOCOS ATRASO : DECLARAÇÃO DA ESTRUTURA : y = g * yl : yl = h * (u - y) :	

segundo bloco de diálogo:

INTRODUÇÃO DOS MODELOS $G = (s^1 - 3.4) / (s^2 + 2 * s^1 - 3.5)$ $H = (3.54) / (s^1 - 5.43) * (s^1 + 2.4)$	
GRAVAR O PROCESSO ? n	

Figura 5 - Blocos de diálogo da declaração.

## 2 - Lugar das raízes

A definição dos dados, para o cálculo do lugar das raízes de um sistema PCl declarado, é realizada da forma apresentada na figura 6 abaixo:

bloco único de diálogo:

NOME : pcl COEFICIENTES VÁRIAVEIS DO CONTROLE BLOCO (G1) - K1 BLOCO (G2) - K2 PARÂMETRO : k1 PARÂMETROS DE CÁLCULO VALORES DE K1 (MIN,MAX): 10,20 VALOR DE K2 : 15.4 NÚMERO DE PONTOS : 120	
(realização dos cálculos)	

Figura 6 - Bloco de diálogo do lugar das raízes.

Obs: a tela de comunicação seguinte corresponde a apresentação do gráfico.

#### 4.6 - Conclusão

Este capítulo introduziu a definição do programa a ser desenvolvido, ressaltando o seu objetivo, as características da declaração do sistema, o conjunto de ferramentas disponíveis, as possibilidades da visualização gráfica dos resultados e, finalmente, os princípios de comunicação utilizados na interface homem-máquina, destacando-se as janelas de comunicação, as formas de escolha das ferramentas, a forma de realização dos diálogos e o editor de textos do programa. A seguir, é apresentada a forma como o programa está estruturado.

## C A P Í T U L O 5

### ESTRUTURA GERAL DO PROGRAMA

Este capítulo descreve a estrutura do programa, apresentando inicialmente uma descrição informal das partes principais deste, e, a seguir, uma descrição formal utilizando a representação em diagramas SADT ("Structured Analysis and Design Technique") (ROSS, 1977) que mostram, através de composições sucessivas, as funções que compõem o programa e suas interrelações.

#### 5.1 - Descrição informal

O programa, cuja estrutura geral é apresentada esquematicamente na figura 1, é composto basicamente por três partes principais: a unidade central de gerenciamento (ou gerenciador), o conjunto de unidades funcionais e a interface homem-máquina. Na figura 1, os traços finos indicam que as várias partes separadas por estes podem, eventualmente, se comunicar diretamente entre si, enquanto que o sombreado indica separação entre as partes, implicando que estas somente são acessadas através do gerenciador.

##### 5.1.1 - Gerenciador

O gerenciador tem como função coordenar a execução do programa através do escalonamento das operações a serem



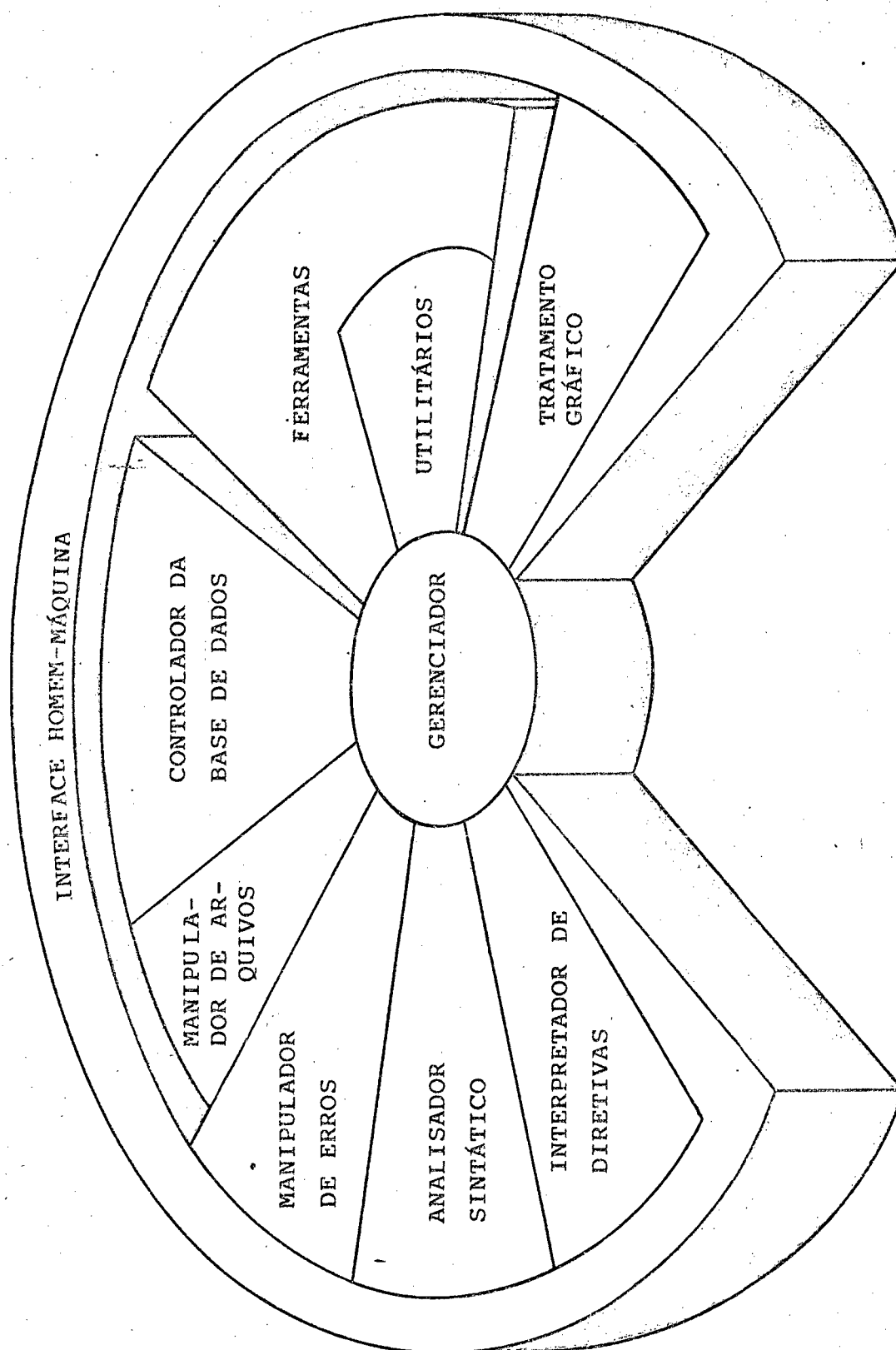


Figura 1 - Estrutura geral do programa

realizadas e, conseqüentemente, o acesso às diversas unidades funcionais. Ele é formado por:

- um alocador dinâmico de memória, o qual gerencia a parte da memória reservada para o armazenamento de dados formais e de resultados;
- um seqüenciador de tarefas, o qual seleciona a tarefa seguinte a ser realizada em função do estado atual de execução do programa e do status da última tarefa realizada;
- um ativador de tarefa, o qual ativa a tarefa anteriormente selecionada.

#### 5.1.2 - Unidades funcionais

O conjunto de unidades funcionais do programa está dividido em módulos que realizam as seguintes operações:

- interpretador de diretivas: identifica a diretiva ou a opção do menu selecionada pelo usuário;
- analisador sintático: analisa os dados introduzidos pelo usuário. Os dados podem ser de três tipos:
  - i) as respostas introduzidas às perguntas apresentadas durante o diálogo de cada ferramenta: neste caso, a análise verifica apenas se o dado é consistente com o tipo de dado esperado (valor numérico, caracter, etc);
  - ii) o conjunto de equações que definem um sistema representado por diagrama de blocos: neste caso a análise é realizada em dois passos: primeiramente é verificada a consistência de cada equação declarada e, após, é verificado se todas as

equações necessárias foram declaradas;

iii) o código correspondente à diretiva: neste caso, é verificado se o código pertence ao conjunto de códigos das diretivas do programa;

- manipulador de erros: permite tratar os erros acontecidos durante a execução do programa, tanto a nível do gerenciador como das unidades funcionais, ativando mensagens de erros e fornecendo auxílio para o usuário;
- manipulador de arquivos: é o responsável por todas as operações de manuseio de arquivos, como por exemplo a criação, a destruição, a leitura do conteúdo, etc;
- controlador da base de dados: é o responsável pela coordenação de todas as operações que envolvam o armazenamento e/ou manipulação e o acesso de dados na memória. Essas operações incluem a montagem e a manipulação da estrutura de dados do sistema, o armazenamento de respostas introduzidas nos diálogos e o armazenamento de resultados intermediários de cálculos;
- ferramentas: contém o conjunto de rotinas disponíveis no programa para análise, projeto e simulação. As ferramentas são ativadas pelo gerenciador a pedido do usuário.
- utilitários: contém o conjunto de rotinas de uso geral, as quais podem ser acessadas tanto pelo gerenciador do programa como pelas ferramentas. Dentro desta classe estão incluídas as rotinas matemáticas (operações com polinômios, operações com matrizes, etc.), os algoritmos de redução de grafos, etc.
- tratamento gráfico: contém todas as rotinas destinadas ao traçado e à manipulação dos gráficos. Essas rotinas são

ativadas pelo gerenciador em duas situações:

- i) a pedido do usuário quando da ativação de uma opção de saída;
- ii) automaticamente, após o uso de uma ferramenta cujos resultados podem ser apresentados na forma gráfica.

### 5.1.3 - Interface homem-máquina

A interface homem-máquina representa o mecanismo através do qual o usuário acessa as ferramentas disponíveis no programa. Ela contém:

- um módulo de tratamento de janelas, o qual prepara a tela de comunicação do programa e coordena o acesso às várias janelas desta;
- um módulo de comunicação por diálogos e menus, o qual é o responsável por todas as operações envolvendo a comunicação com o usuário para a entrada de dados, tanto para a seleção da ferramenta como para a sua utilização;
- os "drivers" de entrada e saída, os quais constituem os acessos aos suportes físicos que permitem se comunicar com o usuário.

## 5.2 - Descrição formal

### 5.2.1 - Objetivo da descrição formal

A representação de um problema na forma narrativa pode incorrer em erros, contradições, repetições, omissões e ambigüidades, além de não apresentar de forma clara a extensão deste. O uso de técnicas de formalização (algébricas, gráficas, etc.), facilitam o entendimento do problema evitando as falhas

anteriormente citadas e permitindo diminuir os riscos para as etapas posteriores do desenvolvimento do software.

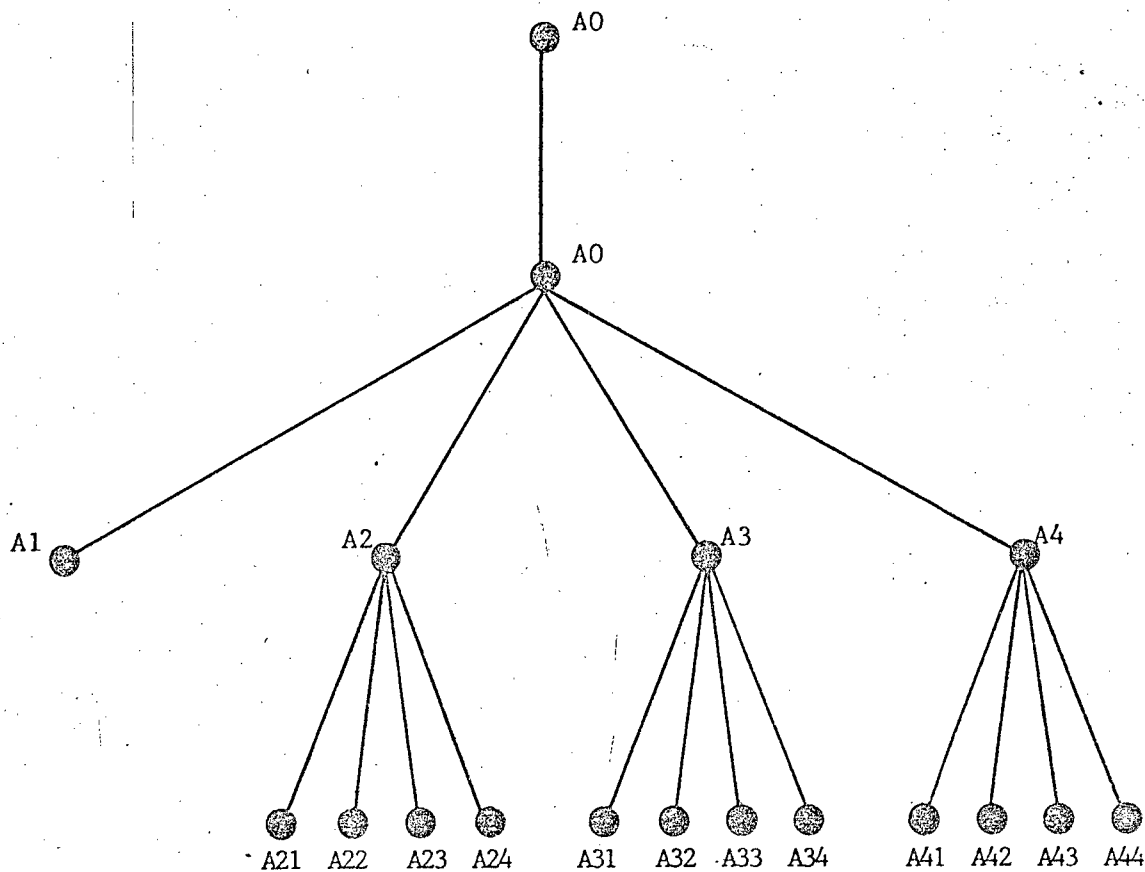
#### 5.2.2 - Metodologia de descrição formal

A metodologia a ser utilizada para representar as características do programa é o SADT ("Structured Analysis and Design Technique") (ROSS, 1977), a qual foi escolhida devido as seguintes razões principais:

- i) pode ser utilizada para representar várias fases do desenvolvimento do software, o que permite minimizar o uso de diferentes metodologias;
- ii) permite uma decomposição do problema em uma forma estruturada de cima para baixo ("top-down");
- iii) tem uma forma de representação gráfica que possui regras de construção bem determinadas, resultando em diagramas bem organizados e fáceis de serem lidos.
- iv) é uma técnica amplamente utilizada mundialmente e aplicável a um grande conjunto de problemas;

As características principais da metodologia SADT são apresentadas resumidamente no anexo 1.

A descrição formal do programa em diagramas SADT, apresentada a seguir, tem como objetivo detalhar as funções principais deste e mostrar como estas se relacionam. Por isso, somente serão apresentados, a seguir, os diagramas correspondentes aos primeiros níveis de decomposição; o primeiro diagrama apresentado mostra a forma com que foram realizadas as decomposições.



A0 : Programa de análise e projeto auxiliado por computador p2  
 para sistemas de controle  
 A1 : Gerenciar programa p3  
 A2 : Gerenciar entrada e saída p4  
     A21 : Realizar comunicação p7  
     A22 : Coordenar base de dados p8  
     A23 : Coordenar arquivos p9  
     A24 : Tratar erros p10  
 A3 : Utilizar ferramentas p5  
     A31 : Utilizar ferramentas clássicas p11  
     A32 : Utilizar ferramentas modernas p12  
     A33 : Simular Sistema p13  
     A34 : Utilizar ferramentas auxiliares p14  
 A4 : Apresentar resultados p6  
     A41 : Apresentar gráficos na tela p15  
     A42 : Apresentar gráficos no plotter p16  
     A43 : Apresentar relatório na impressora p17  
     A44 : Apresentar resultados formais e numéricos p19

LCMI EEL	Nome do Projeto: PACSC		Etapas		Leitor	Data
	Autor(es): VÂNIO M. SAVI		Proposta			
	Data: 28/03/87		Aprovação			
	Versão: 1.3		Publicação		X	

ativação do programa

↓

PROGRAMA PARA ANÁLISE  
E PROJETO AUXILIADO  
POR COMPUTADOR PARA  
SISTEMAS DE CONTROLE

↑

D.O.S

dados de inicialização e configuração do programa

↑

dados do sistema

↑

dados para uso das ferramentas

↑

dados de informação sobre o sistema

↑

resultados

↑

programa configurado

↑

mensagens de erro

↑

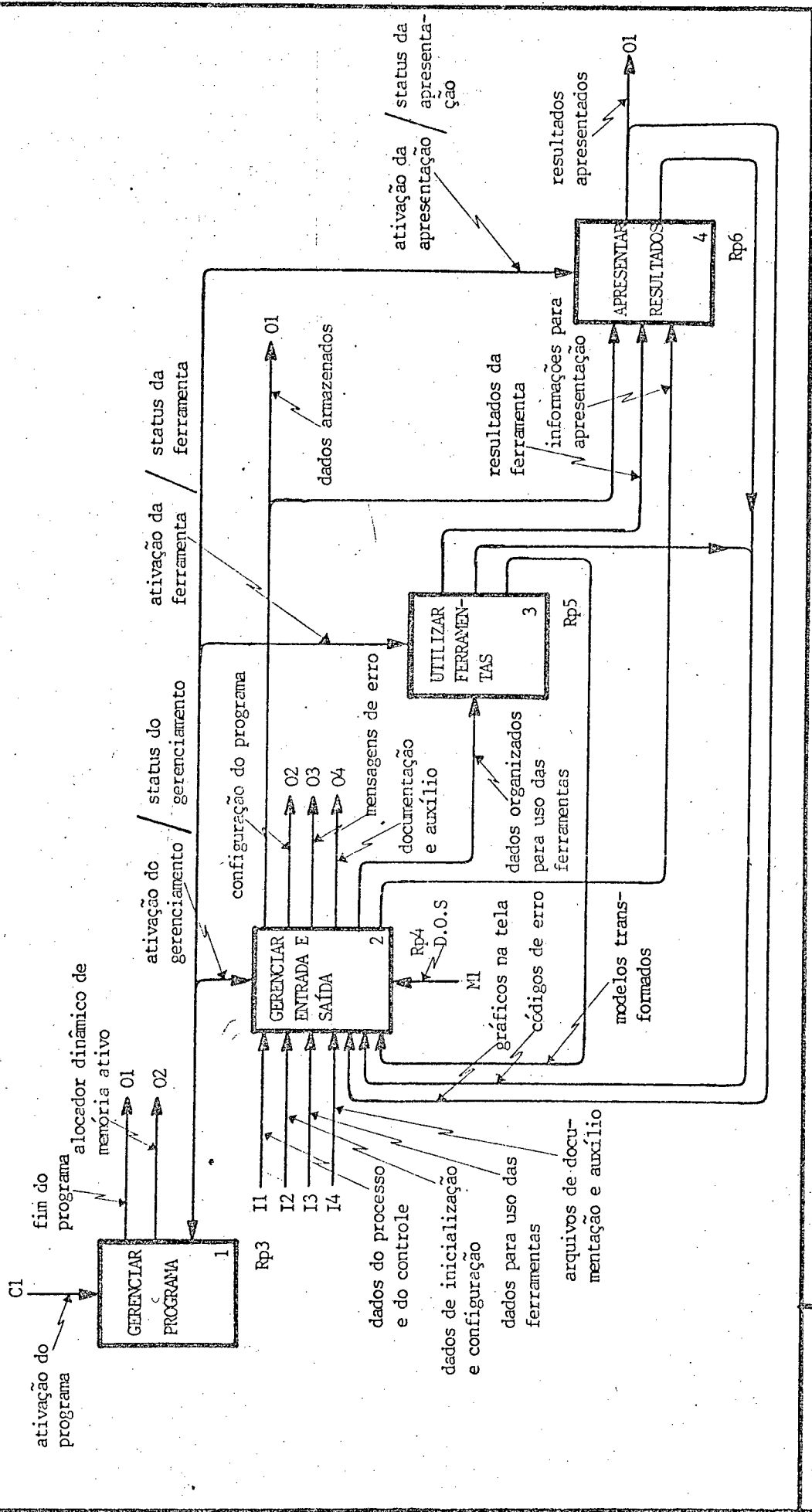
documentação e auxílio

↑

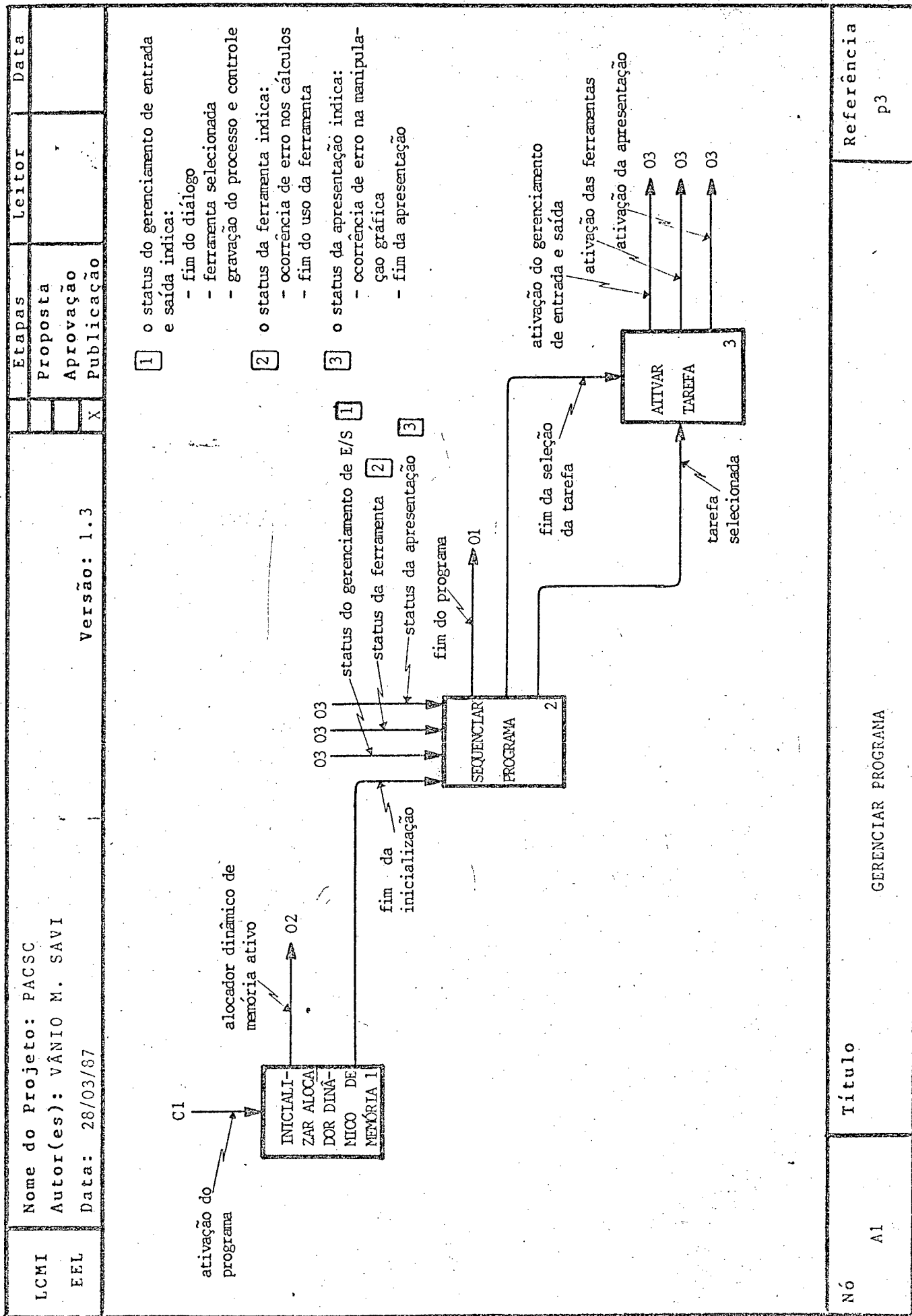
Nó AO	Título	Referência pl
----------	--------	------------------

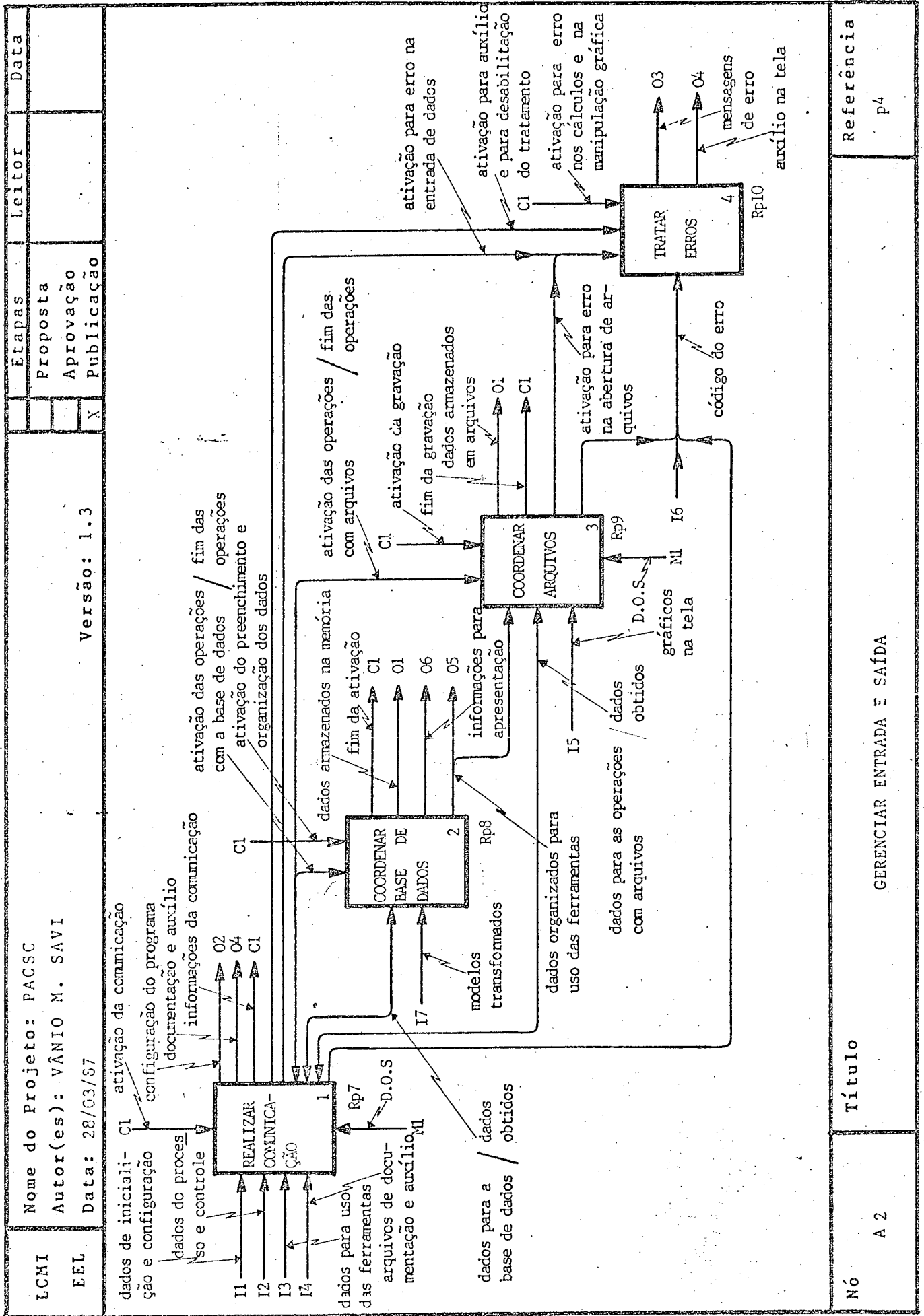
LCMI EEL	Nome do Projeto: PACSC		Etapas		Leitor	Data
	Autor(es): VÂNIO M. SAVI		Proposta			
Data: 28/03/87		Versão: 1.3		Aprovação		
				Publicação	X	



Nó A0	Título PROGRAMA PARA ANÁLISE E PROJETO AUXILIADO POR COMPUTADOR PARA SISTEMAS DE CONTROLE	Referência p2
----------	--	------------------







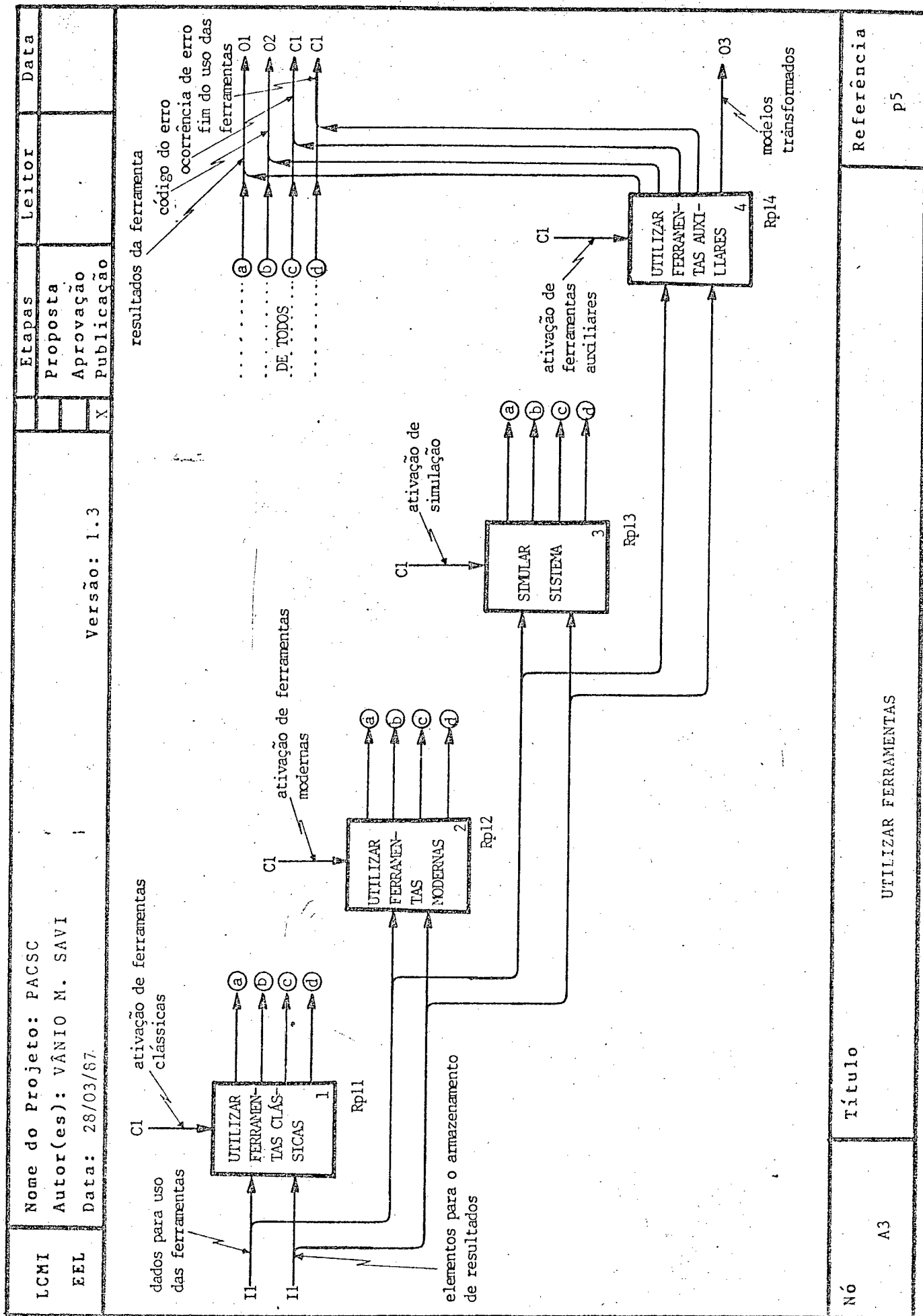
Título

GERENCIAR ENTRADA E SAÍDA

Nº A 2

Referência

p4



Nó

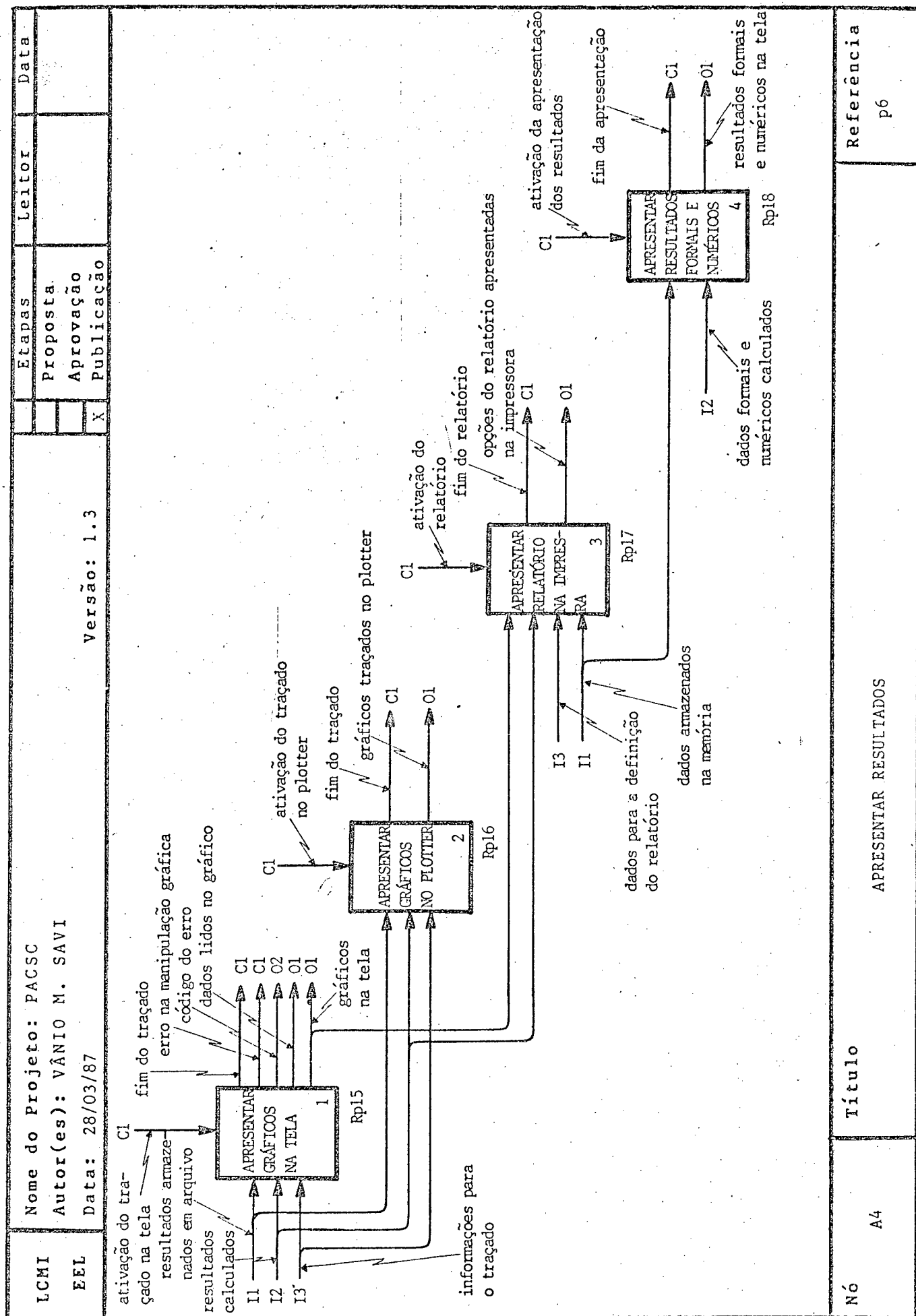
A3

Título

UTILIZAR FERRAMENTAS

Referência

p5



Título

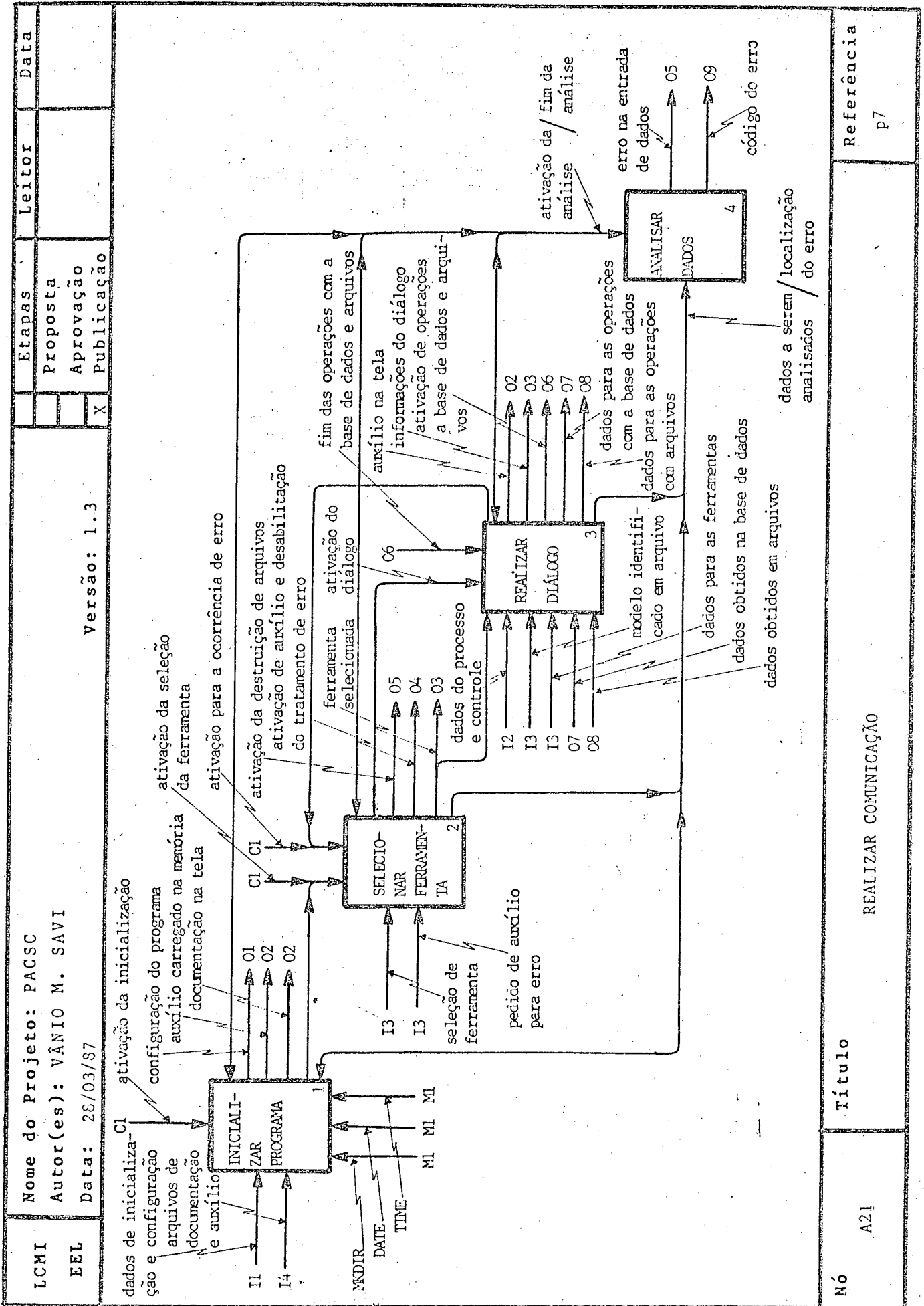
APRESENTAR RESULTADOS

Referência

p6

Nó

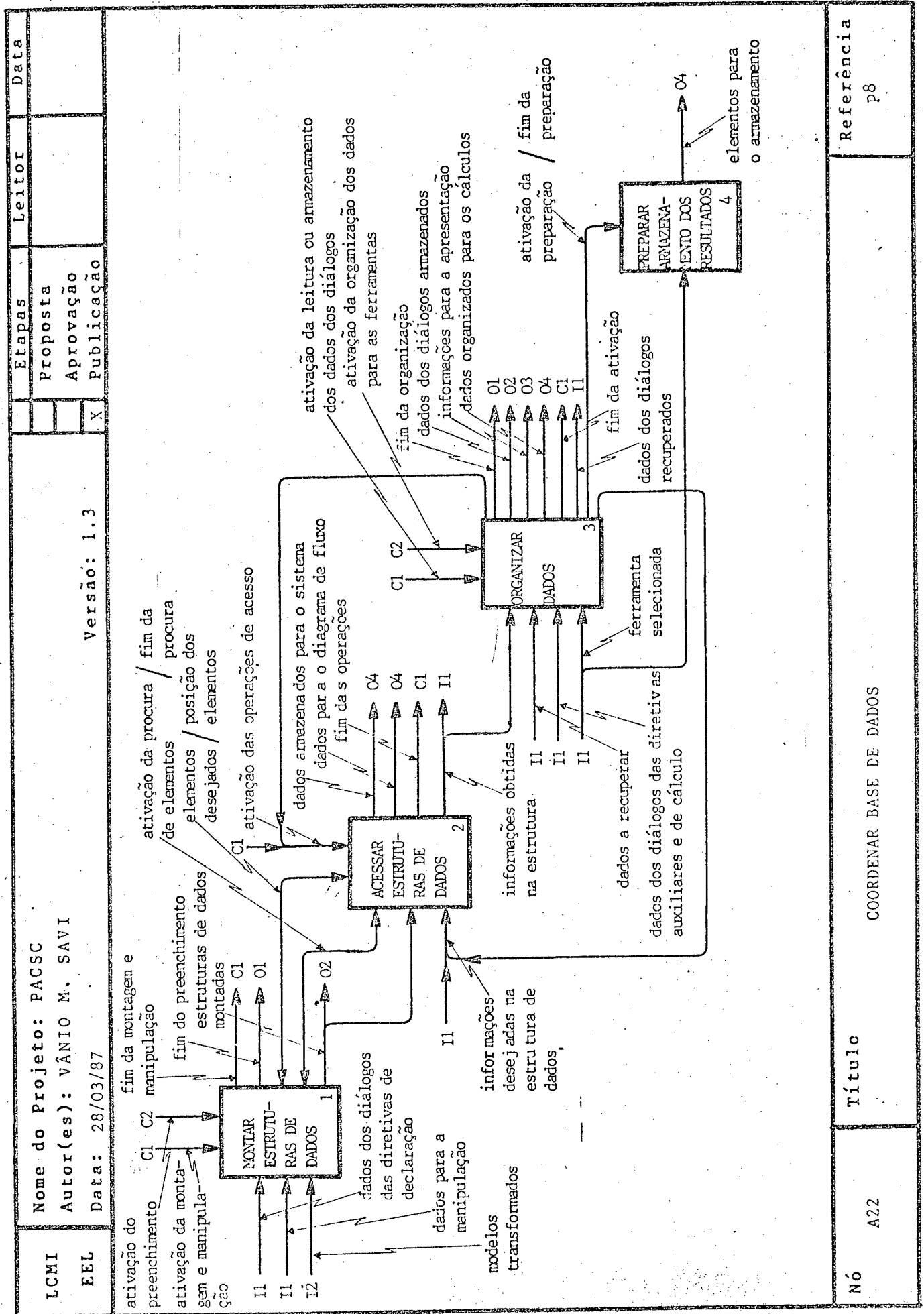
A4

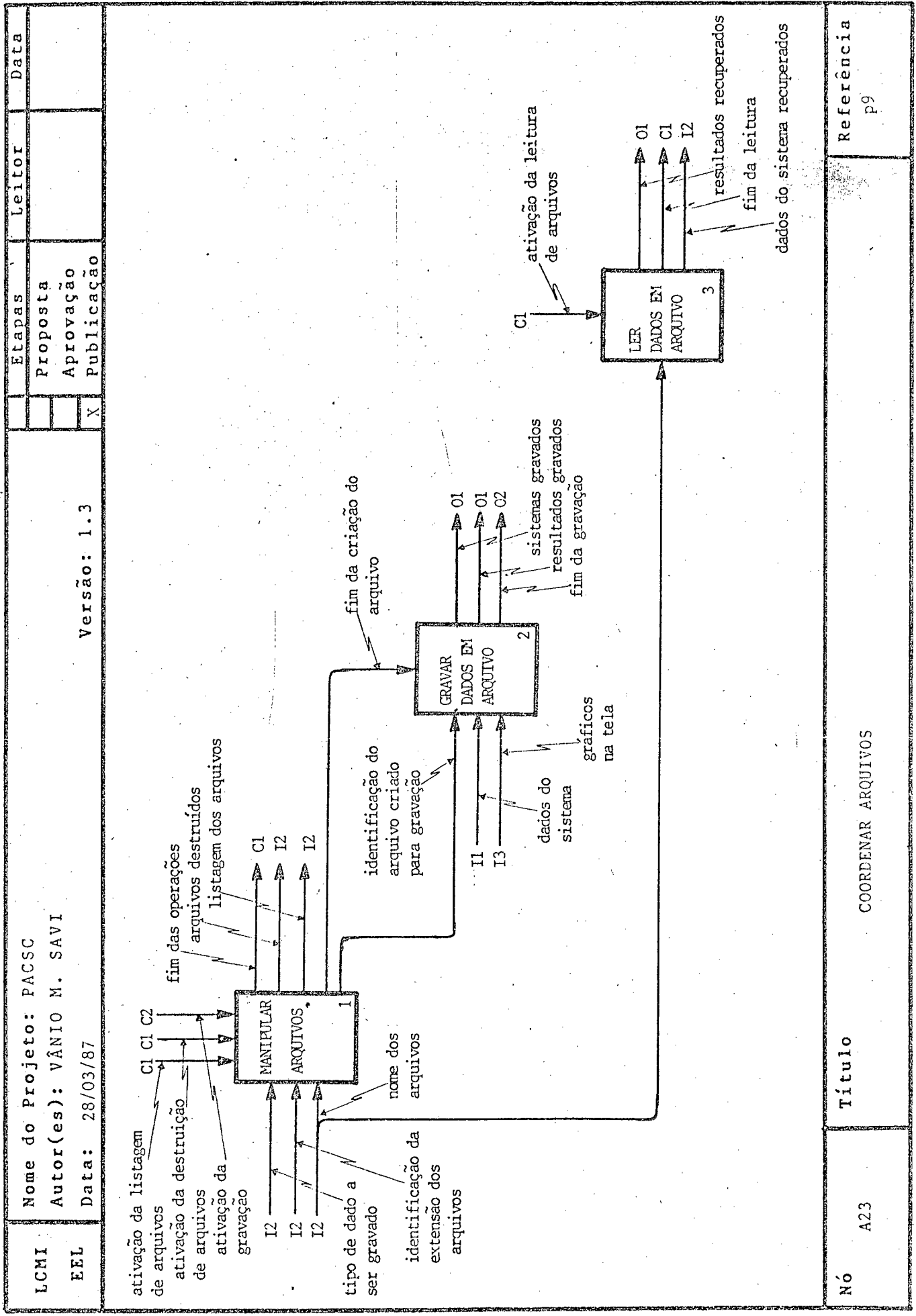


Referência  
p7

Título  
REALIZAR COMUNICAÇÃO

Nó  
A21



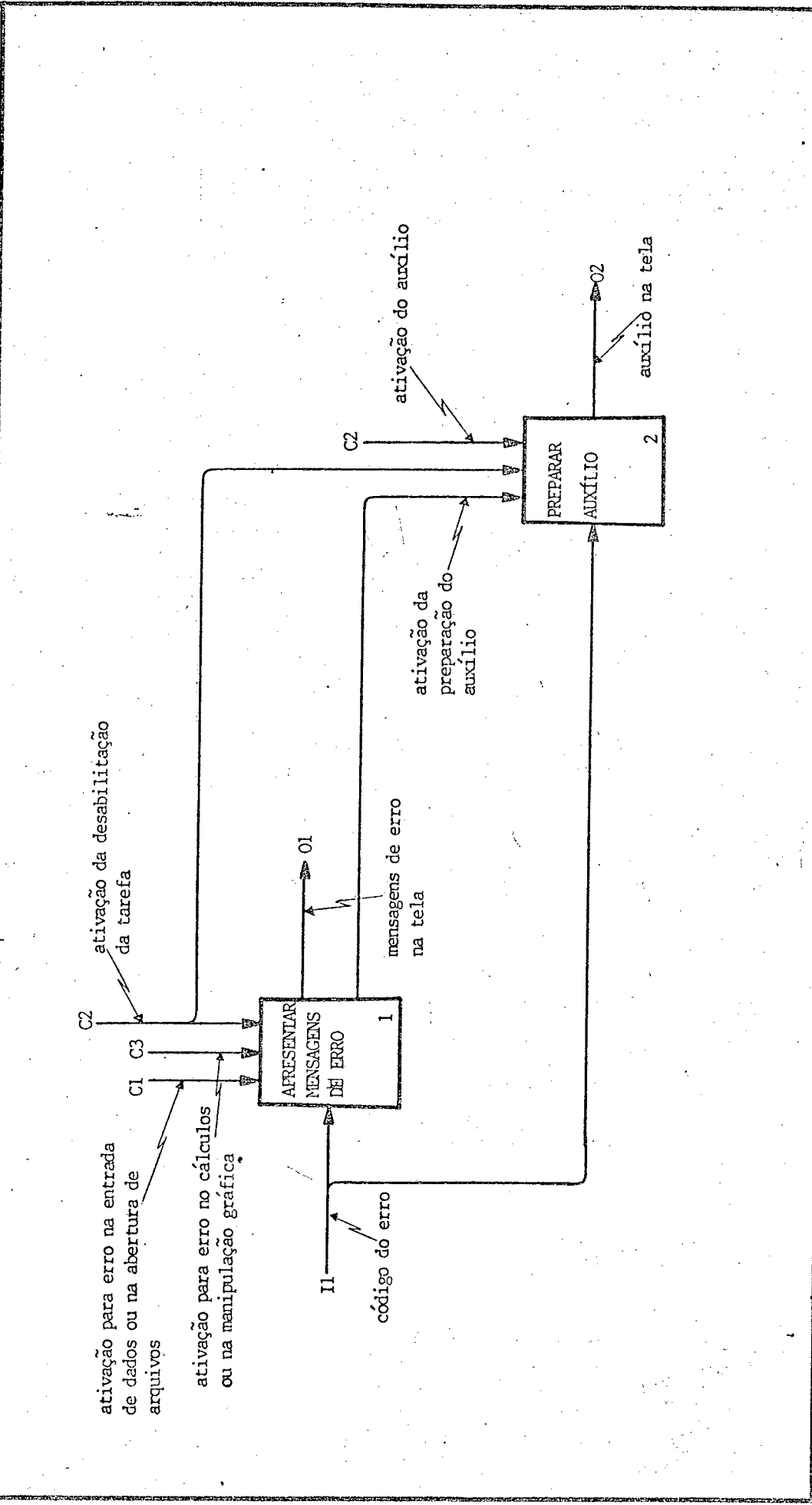


Referência  
p9

Título  
COORDENAR ARQUIVOS

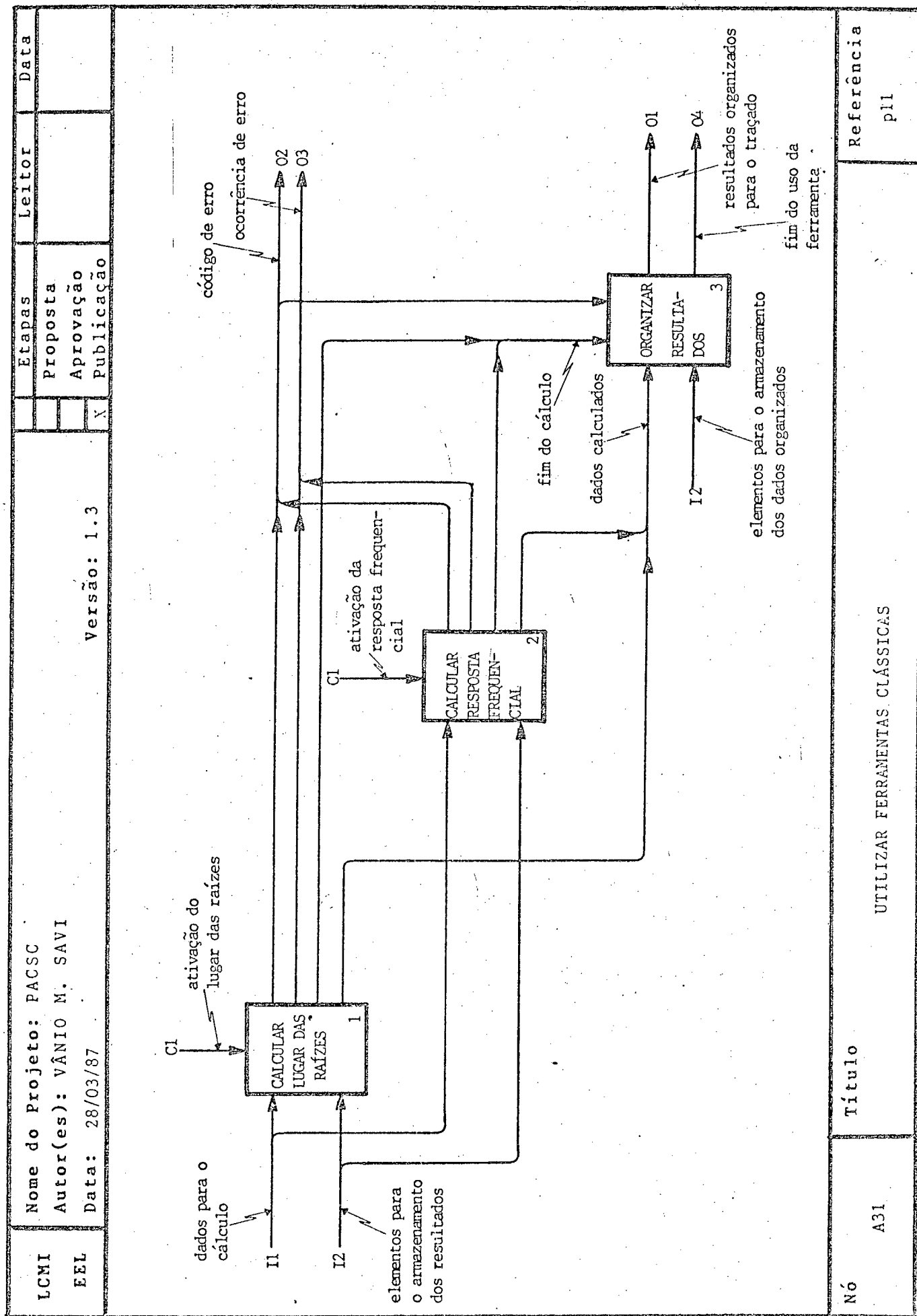
Nó  
A23

LCMI EEL	Nome do Projeto: PACSC		Etapas		Leitor	Data
	Autor(es): VÂNIO M. SAVI		Proposta			
Data: 28/03/87		Versão: 1.3		Aprovação		
				Publicação	X	



Nó A24	Título TRATAR ERROS	Referência p10
-----------	------------------------	-------------------





UTILIZAR FERRAMENTAS CLÁSSICAS

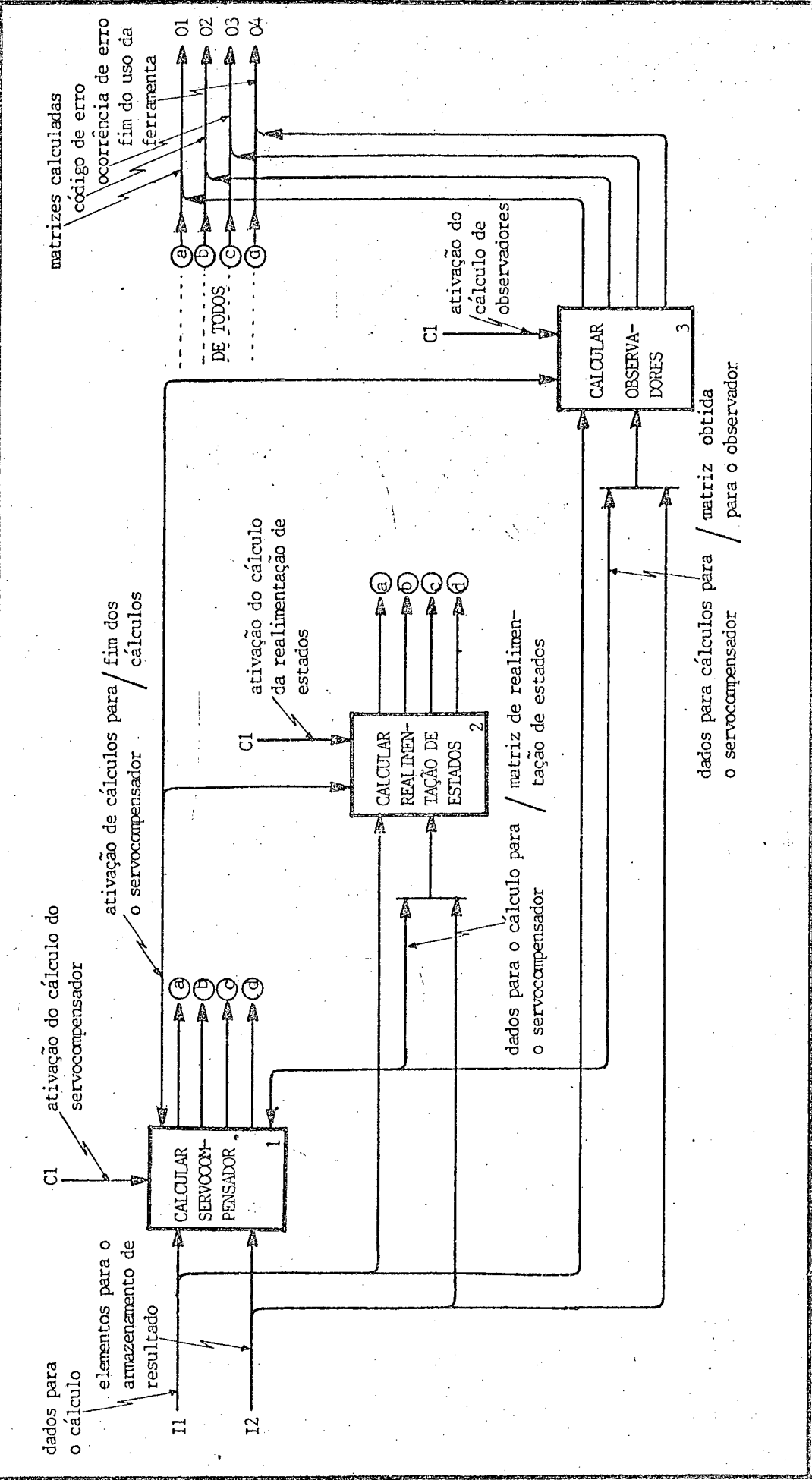
Título

Nó  
A31

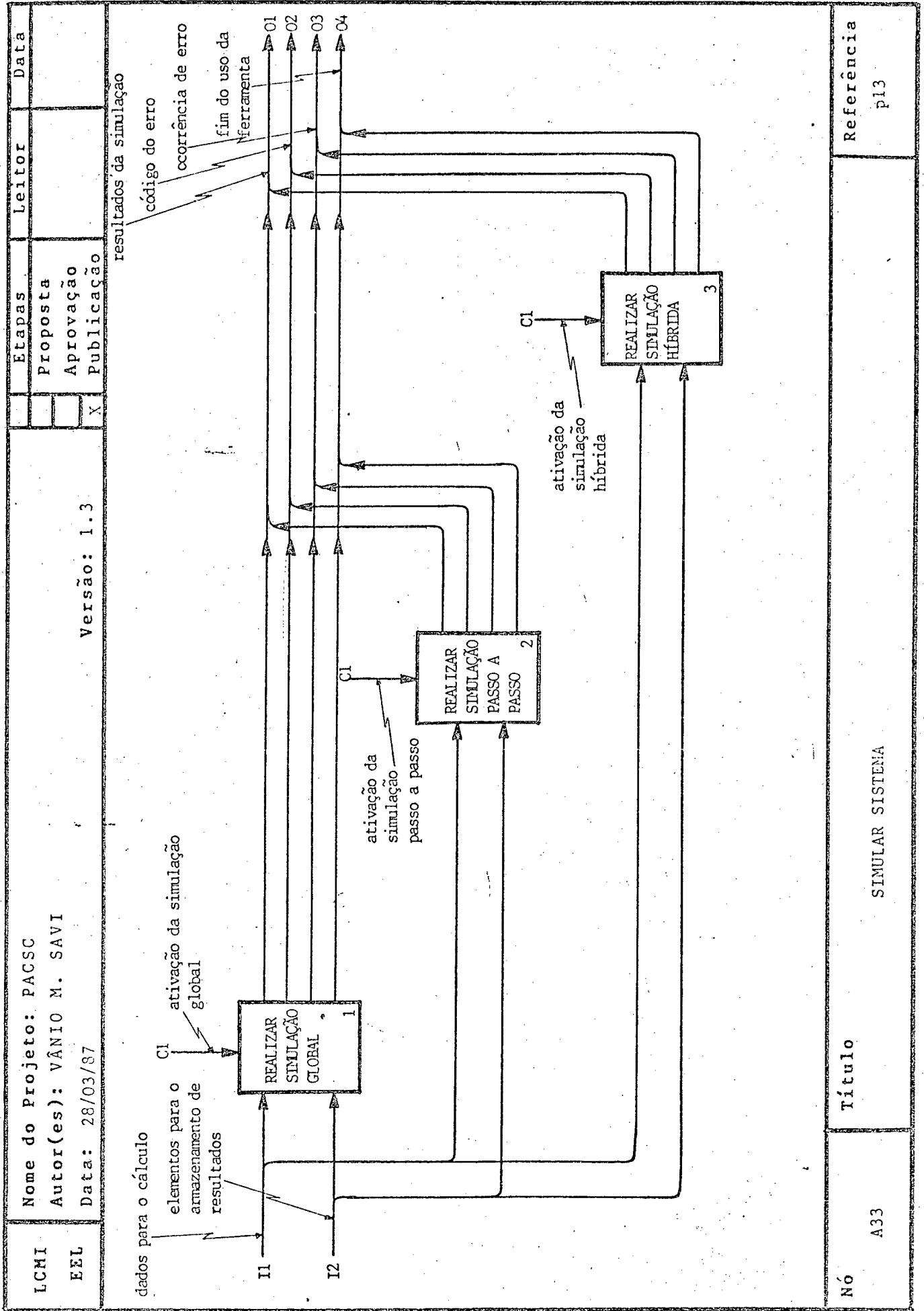
Referência

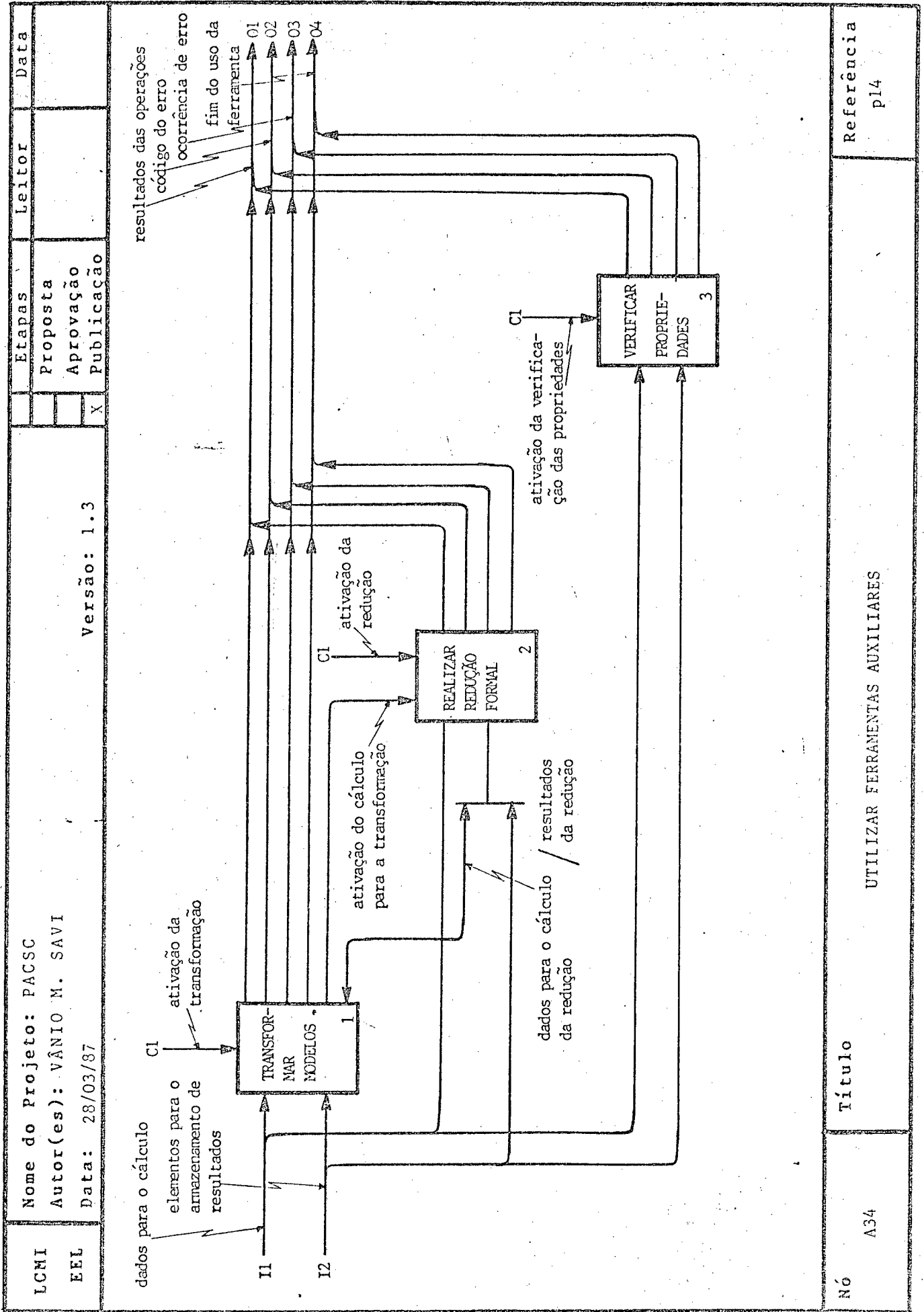
p11

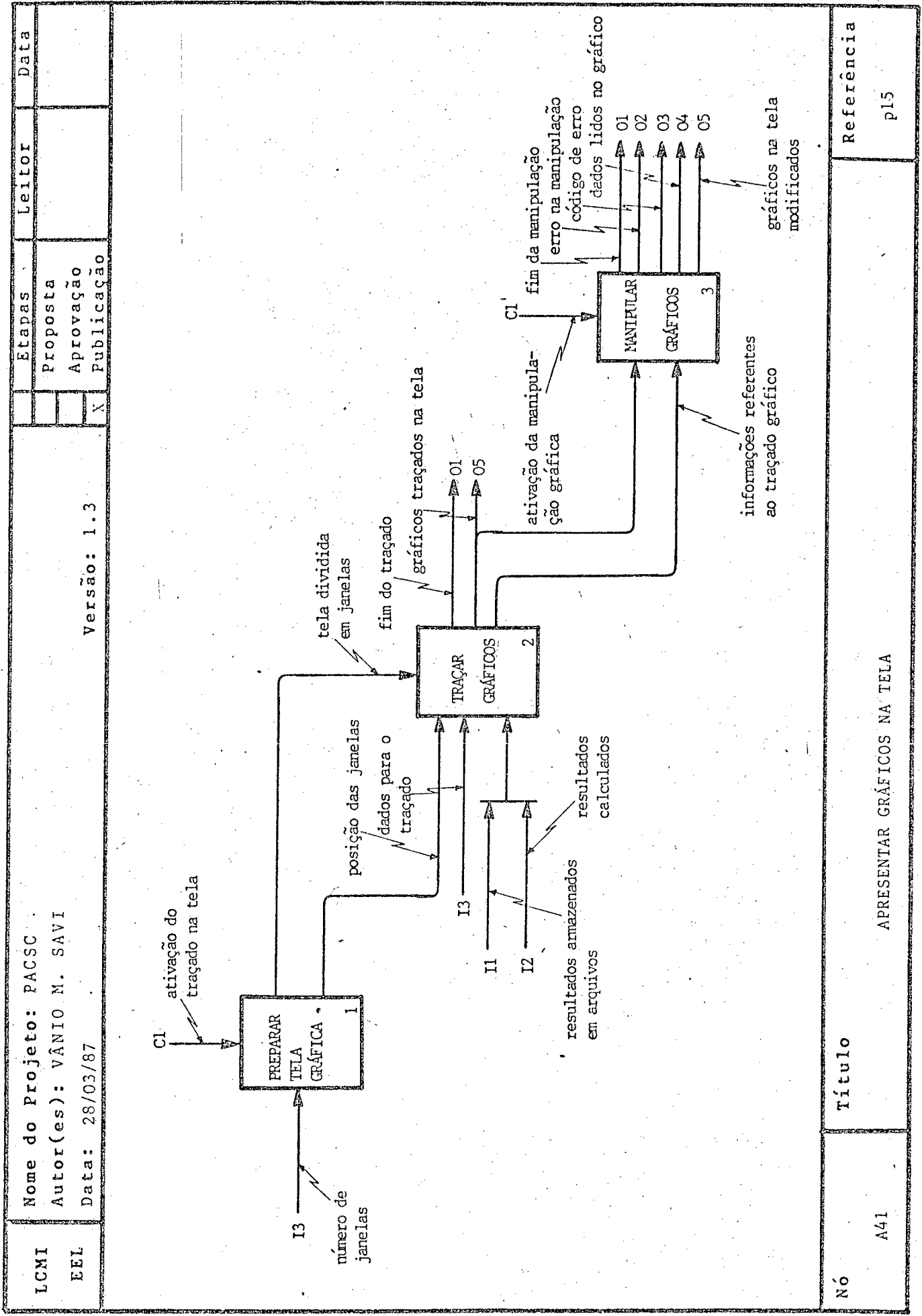
LCMI EEL	Nome do Projeto: PACSC		Etapas		Leitor	Data
	Autor(es): VÂNIO M. SAVI		Proposta			
Data: 28/03/87		Versão: 1.3		Aprovação		
				Publicação	X	

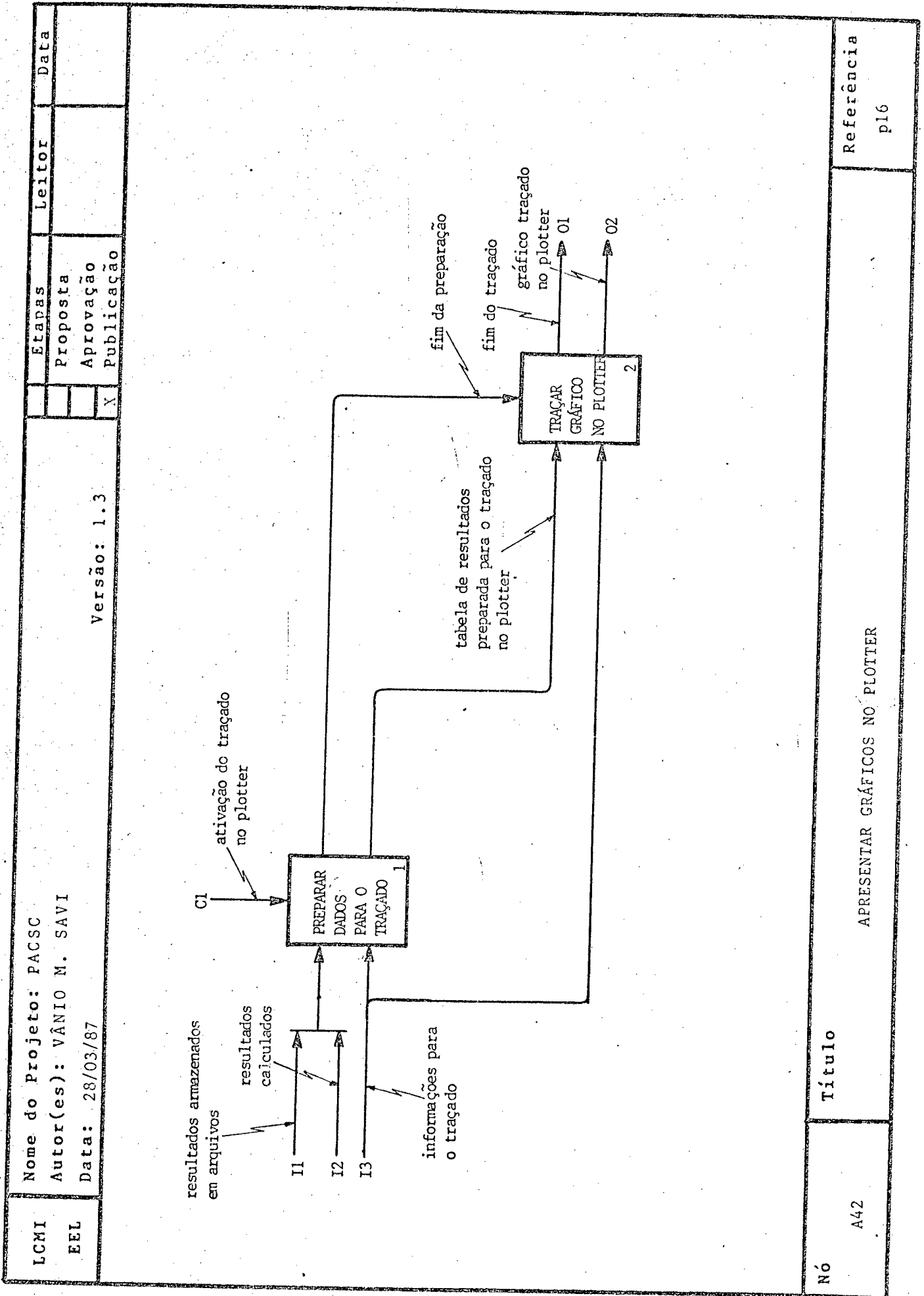


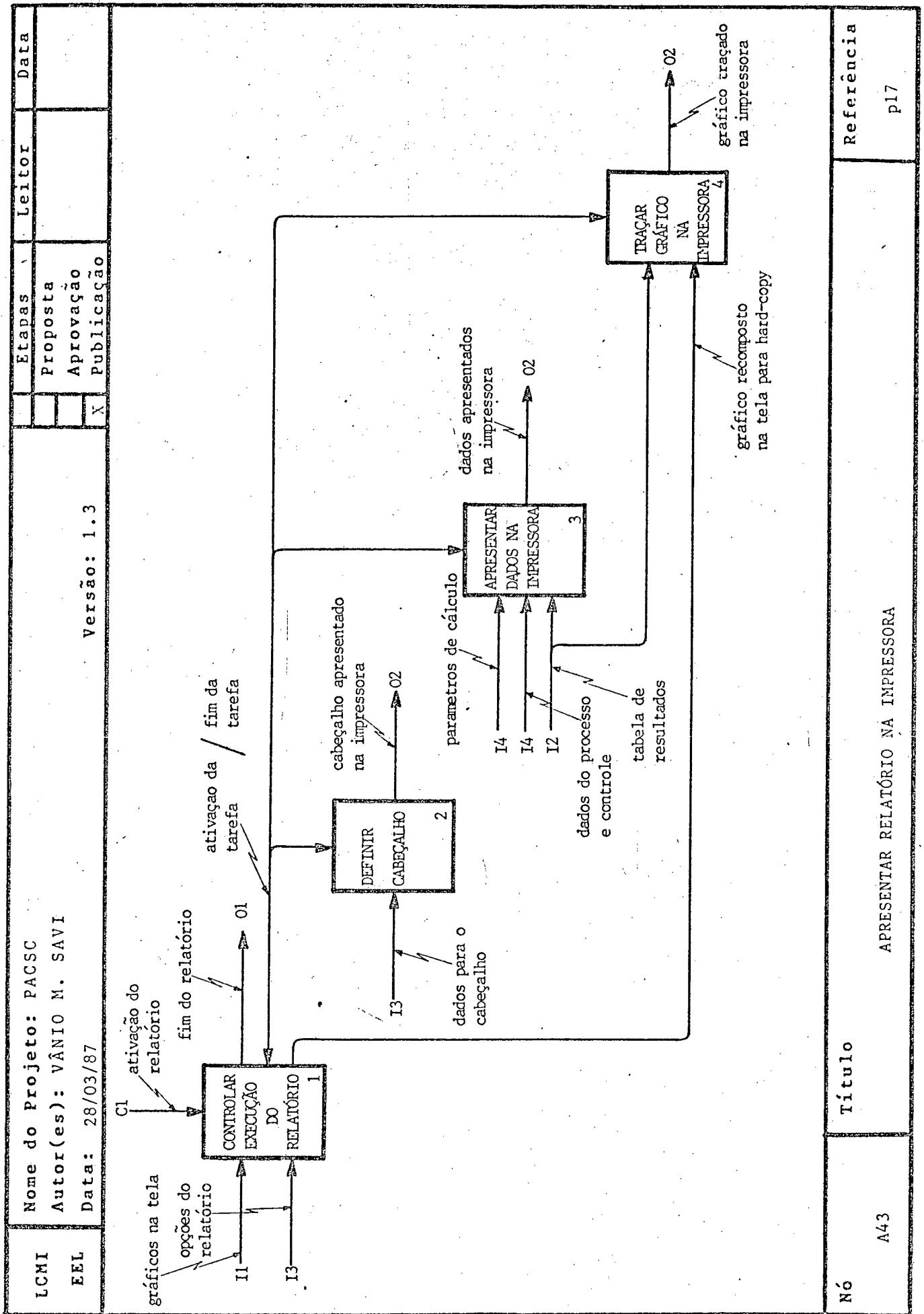
Nó A32	Título UTILIZAR FERRAMENTAS MODERNAS	Referência pl2
-----------	---	-------------------











Título

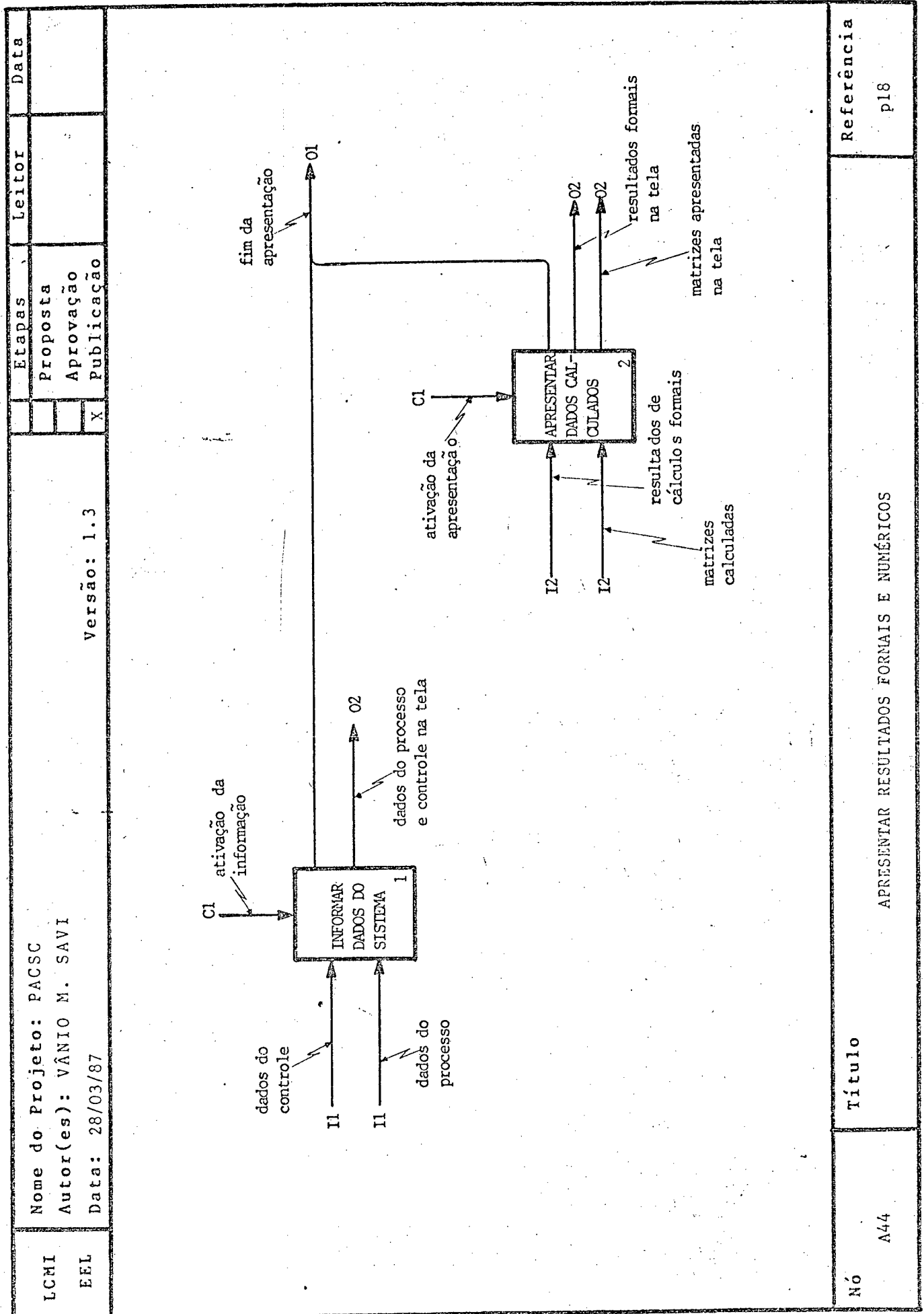
APRESENTAR RELATÓRIO NA IMPRESSORA

Nó

A43

Referência

p17





### 5.3 - Conclusão

Neste capítulo, apresentou-se a estrutura geral do programa, iniciando por uma descrição informal das suas partes principais e, após, formalizando esta descrição na forma de diagramas SADT, que decomporam o programa nas suas funções principais e mostraram a forma como estas se relacionavam entre si. O próximo capítulo destaca a forma com que os dados são armazenados e acessados.

## C A P Í T U L O 6

### ESTRUTURA DE DADOS

#### 6.1 - Introdução

O armazenamento de dados (formais e numéricos) na memória é organizado na Base de Dados do programa, a qual utiliza para este fim um bloco pré-especificado da memória; este bloco é gerenciado por um alocador dinâmico de memória associado ao gerenciador do programa.

As estruturas utilizadas para o armazenamento dos dados dependem do tipo de dados e das operações a serem realizadas sobre estes. Basicamente, três tipos de estruturas formam a base de dados:

- i) a estrutura para o armazenamento dos resultados: esta estrutura é formada por elementos do tipo matrizes e tabelas criados dinamicamente de acordo com as necessidades de armazenamento de cada ferramenta; estes elementos são destruídos após a apresentação dos resultados;
- ii) a estrutura para o armazenamento dos dados do diálogo das diretivas de cálculo (simulação, resposta em frequência e lugar das raízes): esta estrutura utiliza elementos do tipo registros para armazenar as respostas

introduzidas no diálogo; esses elementos, uma vez criados, não são mais destruídos;

- iii) a estrutura para o armazenamento dos modelos do sistema (processo e controle): esta estrutura utiliza para o armazenamento dos dados um conjunto de elementos do tipo registros interligados na forma de uma árvore binária.

Neste capítulo, serão apresentadas apenas as características desta última estrutura de dados, por esta ser a principal estrutura de armazenamento de dados e por possuir um maior grau de complexidade em relação às anteriores, as quais são bastante comuns. Assim, será descrita, a seguir, a forma com que a estrutura de dados do sistema é montada e acessada.

#### 6.2 - Objetivo de uma estrutura para o armazenamento dos modelos do sistema

O armazenamento, numa forma estruturada dos dados declarados pelo usuário para o sistema (processo e controle), é de fundamental importância para o bom desempenho do programa, principalmente para a leitura ou modificação dos dados armazenados. Por isso, deve ser definida uma estrutura de dados apropriada para o armazenamento dos dados relativos à descrição formal e numérica do sistema, assim como das outras representações obtidas internamente pelo programa através das transformações de modelo.

A estrutura de dados deve satisfazer os seguintes requisitos principais:

- i) permitir o fácil acesso a todos elementos da estrutura

- para leitura, ou modificação, dos seus respectivos dados;
- ii) otimizar a memória necessária para o armazenamento dos dados de cada sistema declarado;
  - iii) possibilitar a montagem e modificação da estrutura de forma dinâmica;
  - iv) possibilitar o armazenamento das equações declaradas para os modelos dos sistemas;
  - v) possibilitar o armazenamento para cada elemento do sistema (variáveis, blocos) dos dados necessários à sua definição.

### 6.3 - Estrutura escolhida

Dentre os principais tipos de estruturas de dados encontradas na literatura (apresentadas de forma resumida no anexo 2), escolheu-se para o armazenamento dos dados do sistema uma estrutura do tipo árvore binária. A escolha desta estrutura foi baseada no fato desta satisfazer os requisitos apresentados acima, em particular no que concerne a:

- i) armazenamento: cada elemento da árvore pode ser representado por um registro permitindo o armazenamento de diferentes tipos de informações simultaneamente. As equações podem ser armazenadas através da transformação destas em um conjunto de relações do tipo: (variável) = (operando esquerdo) (operador) (operando direito), para as quais a árvore binária é a estrutura de armazenamento mais adequada;
- ii) montagem e acesso: os elementos podem ser, na medida do necessário, criados dinamicamente, preenchidos e interligados através de apontadores. O acesso aos

elementos é realizado através dos clássicos algoritmos de atravessamento da árvore binária (HOROWITZ e SHANI, 1984) que apresentam um bom desempenho;

iii) redução: a árvore binária possibilita a obtenção de forma simples do diagrama de fluxo de sinais, a partir do qual a técnica de redução para a obtenção da função de transferência global é aplicada.

#### 6.4 - Descrição da estrutura de dados

##### 6.4.1 - características construtivas

A árvore binária é formada por diferentes tipos de elementos (nós), os quais são interligados de forma a representar o sistema declarado através de apontadores.

Cada elemento é representado internamente por uma estrutura do tipo registro, possuindo campos específicos segundo o elemento utilizado. Os tipos de elementos a serem utilizados na montagem da árvore e seus respectivos campos são apresentados a seguir; como ilustração, é apresentada a representação esquemática de cada elemento na forma de blocos.

##### - Elemento RAIZ:

Este elemento armazena informações como o código do usuário (COD), o nome do processo e do controle (NP, NC), o tipo de representação do processo e controle (TP, TC) e o código de identificação do controle (CODC); ele contém também os apontadores para as tabelas de entrada, saída e perturbação do processo e do sistema/controle (APTE P, APTS P, APTP P, APTE S/C, APTS S/C, APTP S), e os apontadores para os blocos que armazenam

a representação do processo e do controle por equações de estado (APEE P, APEE C). O elemento raiz está sempre presente na estrutura e sua representação esquemática é apresentada na figura 1.

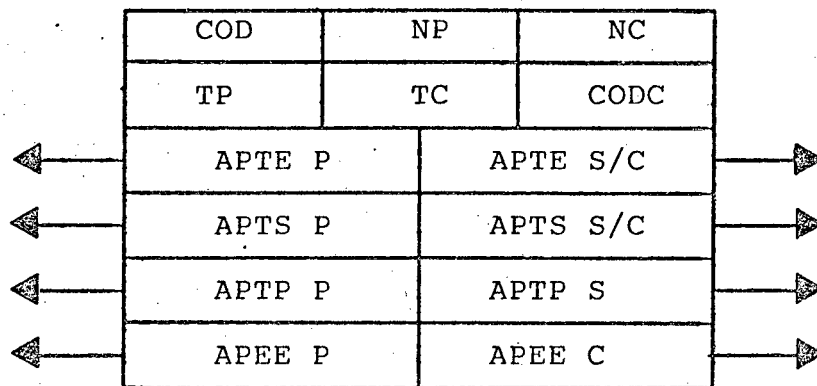


Figura 1 - Elemento raiz

- Elemento VARIÁVEL:

Este elemento armazena informações como o índice do elemento (IND), o nome da variável (NOME), o operador da relação (OP), a dimensão da variável (DIME), os códigos de controle dos apontadores esquerdo e direito (CE, CD); ele contém também os apontadores para os elementos a esquerda e a direita na subárvore (AP\_ESQ, AP\_DIR). O nome da variável, no caso de ter sido criada pelo programa (variável implícita), é formado pelo caracter "\$" seguido de um número inteiro. A representação esquemática deste elemento é apresentada na figura 2.

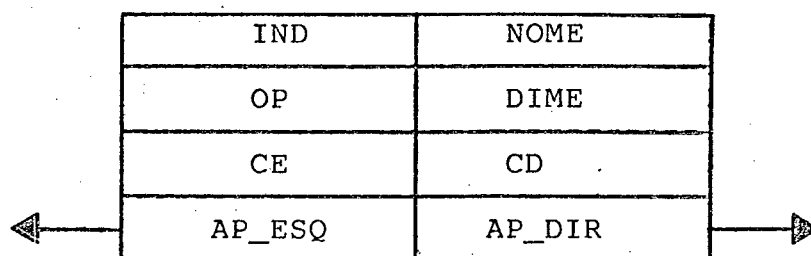


Figura 2.- Elemento variável

- Elemento BLOCO:

As informações armazenadas neste elemento dependem do tipo de representação do bloco.

No caso de representação por funções de transferência, este elemento armazena informações como o índice do elemento (IND), o nome do bloco (NOME), o identificador da origem do bloco (IOB), o identificador do tipo do bloco (TDB), a definição do modelo do bloco (DEF), os parâmetros definidos no modelo dos blocos lineares (PAR1, PAR2, PAR3), os coeficientes dos modelos dos blocos não-lineares ou atraso (M1, M2, M3), o grau do numerador e do denominador do polinômio que representa o modelo (GN, GD) e o apontador deste modelo (AP\_POL), o número de linhas e colunas de cada matriz da representação equivalente do bloco por equações de estado (NL, NC) e os apontadores dos modelos das respectivas matrizes (AP\_MA, AP\_MB, AP\_MC, AP\_MD). Os modelos dos polinômios e matrizes são armazenados em listas encadeadas, separando-se o numerador do denominador no primeiro caso e as linhas no segundo. A representação esquemática deste elemento é apresentada na figura 3.

No caso de representação por equações de estado, este elemento armazena informações como o índice do elemento (IND), o nome do bloco (NOME), a dimensão dos vetores de entrada, saída, perturbação e estado do bloco (DIM E, DIM S, DIM P, DIM X), no caso de blocos monovariáveis o grau do numerador e do denominador da representação equivalente por função de transferência (GN, GD) e o apontador deste modelo (AP\_POL), o número de linhas e colunas de cada matriz da representação do bloco por equações de estado

(NL, NC) e os apontadores dos modelos das respectivas matrizes (AP\_MA, AP\_MB, AP\_MC, AP\_MD, AP\_ME, AP\_MF). A representação esquemática deste elemento é apresentada na figura 4.

IND		NOME	
IOB	TDB	DEF	
PAR1	PAR2	PAR3	
M1	M2	M3	
GN	GD	AP_POL	
NL	NC	AP_MA	
NL	NC	AP_MB	
NL	NC	AP_MC	
NL	NC	AP_MD	




figura 3 - elemento bloco - função de transferência

IND		NOME	
DIM E	DIM S	DIM P	DIM X
GN	GD	AP_POL	
NL	NC	AP_MA	
NL	NC	AP_MB	
NL	NC	AP_MC	
NL	NC	AP_MD	
NL	NC	AP_ME	
NL	NC	AP_MF	




figura 4 - elemento bloco - equações de estado

- Elemento TABELA:

Este elemento é composto de um conjunto de apontadores (AP1,



AP2, AP3, AP4, AP5) que indicam a posição de um determinado tipo de variável na estrutura (entradas, saídas ou perturbações). A sua representação é apresentada na figura 5.

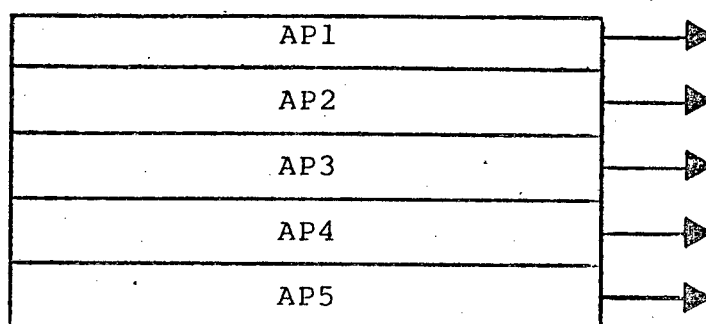


Figura 5 - Elemento tabela.

#### 6.4.2 - Montagem da árvore binária

O processo de montagem da árvore binária é realizado paralelamente à declaração do modelo do processo ou do controle, após o fim da análise de cada bloco de diálogo. Ele inicia pela criação e preenchimento do elemento raiz e pela criação dos elementos tabela necessários; os apontadores destes elementos serão atualizados durante a montagem da parte da estrutura referente às equações declaradas.

A seguir, a partir do conjunto de equações declaradas para o modelo, é completada a estrutura pela criação, interligação e preenchimento parcial dos demais elementos. O preenchimento dos elementos é completado após a definição dos modelos dos blocos ou matrizes. A montagem implica na realização de três etapas:

i) a transformação e o armazenamento das equações:

Nesta etapa, cada equação é transformada para a forma pós-fixada (KRUSE, 1984) inversa, a qual permite a eliminação de

eventuais parênteses da equação. Em seguida, a equação é armazenada em uma lista encadeada cujos elementos tem a seguinte forma:



onde,

Vetor 1 : armazena os componentes da equação (operandos e operadores);

Vetor 2 : armazena o código associado a cada componente. Este código permite identificar se os operandos já apareceram em equações anteriores ou na própria equação (código = 1), se os operandos apareceram nas equações do processo no caso de se tratar das equações do controle (código = 2), ou se ainda não apareceram (código = 0). Nesta etapa os códigos são nulos;

ii) a ordenação das equações:

Nesta etapa, as equações armazenadas na lista são ordenadas a partir de uma variável de saída, de forma a permitir que após a montagem da estrutura todos os elementos possam ser acessados a partir dos elementos que representem estas variáveis. Ao mesmo tempo é realizada a atualização dos códigos correspondentes aos operandos da equação que estiver sendo ordenada na lista;

iii) a montagem propriamente dita:

A partir do conjunto de equações ordenadas, é completada, então, a estrutura em árvore. O algoritmo de montagem (IE BARON,

1979) basicamente se resume em:

a) ler cada equação, componente por componente, a partir do operador de igualdade ("=");

b) identificar o tipo do componente:

b.1) se for um operador, criar um elemento do tipo variável e seguir pelo apontador direito do elemento.

O nome do elemento será preenchido com o rótulo da variável à esquerda da equação, no caso de se tratar do primeiro operador, e com o rótulo correspondente a uma variável implícita nos demais casos;

b.2) se for uma variável ou bloco verificar o código associado:

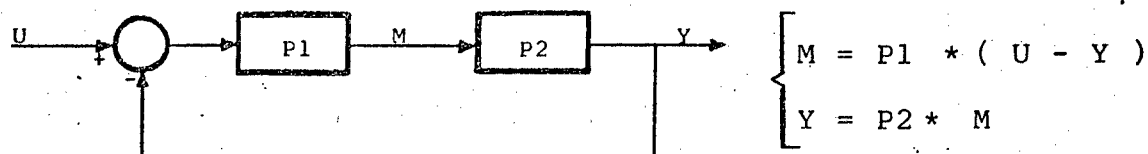
1) se código = 0, criar o respectivo elemento;

2) se código = 1 ou 2, procurar o elemento na estrutura e apontar para ele.

Após isso, voltar ao elemento anterior e seguir pelo apontador esquerdo deste, repetindo o processo;

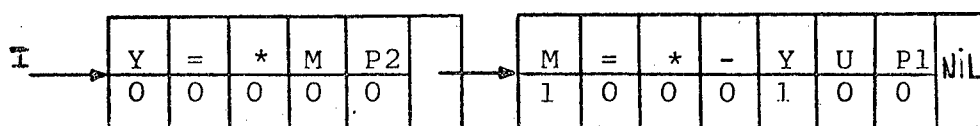
c) ao terminar a equação repetir o processo para as demais equações, verificando primeiramente se a variável à esquerda da nova equação já apareceu em equações anteriores. Caso afirmativo, procurar o elemento na estrutura iniciando a montagem a partir deste.

A figura 6 ilustra de forma simplificada a montagem da estrutura de dados para um sistema declarado através de duas equações e apresenta a forma da estrutura após a inserção de cada elemento de acordo com o algoritmo citado acima.

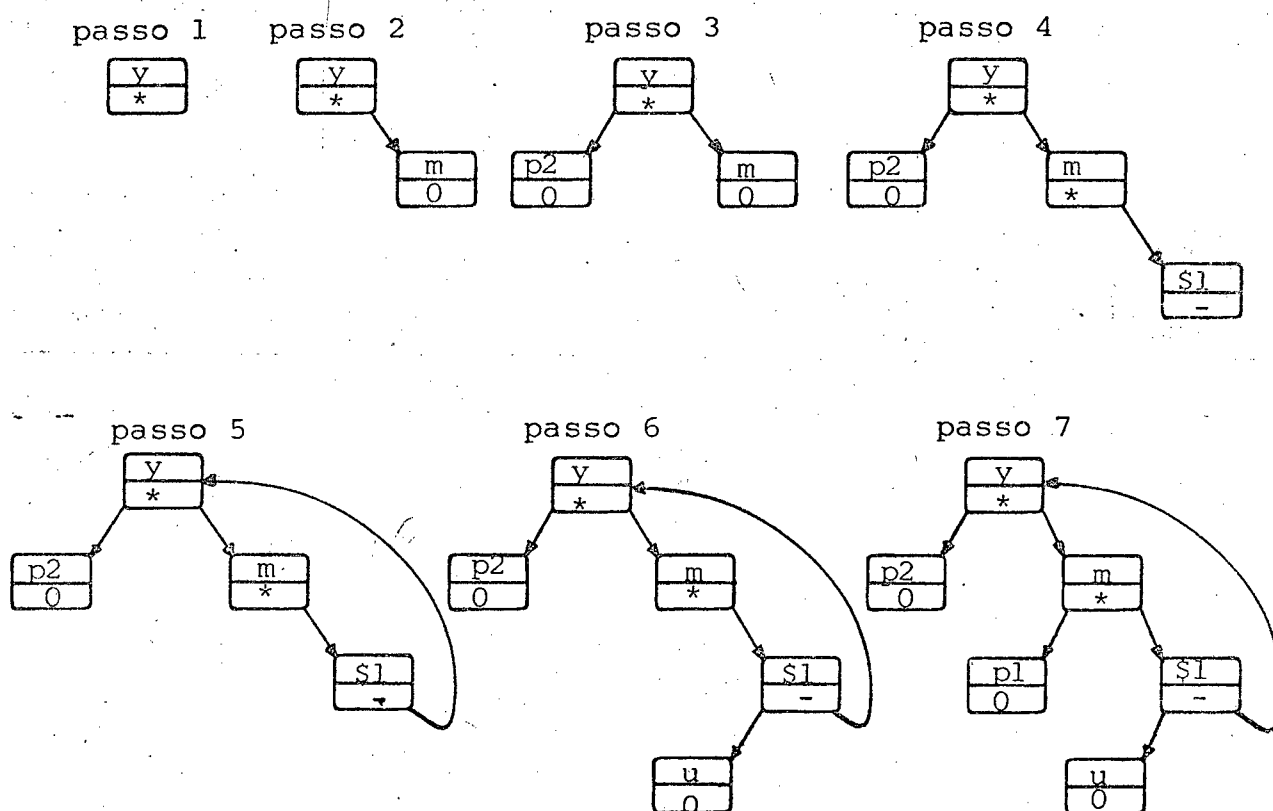


a) diagrama de blocos

b) equações declaradas pelo usuário



c) equações transformadas e ordenadas



d) forma simplificada da estrutura após a inserção de cada elemento

Figura 6.- Exemplo da montagem da estrutura de dados

### 6.4.3 - Tipos de estruturas de dados criadas

Durante a execução do programa, várias estruturas de dados podem ser criadas simultaneamente, em função dos modelos declarados para o processo e o controle e do número de controles declarados. Dois tipos de estruturas são possíveis: a estrutura de dados do sistema e a estrutura de dados do controle.

#### a - Estrutura de dados do sistema

A estrutura de dados do sistema, a qual é formada inicialmente pela estrutura do processo e completada após a declaração do controle, pode ser composta de duas estruturas:

##### a.1 - Estrutura com modelos declarados

Esta estrutura estará sempre presente e corresponde a estrutura montada a partir dos modelos do processo e controle declarados pelo usuário. Ela é a principal estrutura do programa e sobre a qual são realizadas a maioria das operações que envolvem o acesso aos dados armazenados. Esta estrutura armazena, além dos dados declarados pelo usuário, as representações obtidas internamente pelo programa, tais como as representações equivalentes dos modelos de cada bloco do sistema.

##### a.2 - Estrutura global

Esta estrutura é obtida internamente pelo programa, sendo criada nos seguintes casos: sistemas mono e multivariáveis representados por equações de estado; sistemas monovariáveis representados por funções de transferência e que não possuam blocos não-lineares ou atrasos, perturbações ou parâmetros. Ela

armazena o modelo do sistema reduzido a um único bloco representado por equações de estado, permitindo com isso a realização imediata da simulação global do sistema.

#### b - Estrutura de dados do controle

Estas estruturas têm como objetivo armazenar os modelos dos controles declarados que não estão sendo utilizados na formação do sistema. Podem coexistir simultaneamente até quatro estruturas independentes de controle, uma delas estando sempre conectada ao processo. Quando necessário, e de forma simples, estas estruturas podem ser conectadas e desconectadas da estrutura do processo, como será apresentado a seguir.

### 6.5 - Modificação da estrutura de dados do sistema

A modificação da estrutura do sistema substituindo a atual estrutura de controle por outra é realizada de acordo com as seguintes etapas:

i) a criação de uma estrutura para armazenar o controle atual, desconectando-o da estrutura do sistema. Isto é realizado através dos seguintes passos:

a) identificar os elementos do processo, com exceção das entradas, que fazem parte também da estrutura do controle. Ao encontrar estes elementos realizar as seguintes operações:

- criar um elemento auxiliar idêntico ao apontado no processo;
- preencher os campos de identificação do elemento auxiliar (IND, NOME) com os mesmos dados do

elemento apontado no processo;

- mudar o apontador para o elemento auxiliar;

- b) criar os elementos raiz, tabela de entradas e tabela de saídas, preenchendo-os com as informações obtidas na estrutura do sistema;
- c) destruir as informações referentes à estrutura de controle no elemento raiz da estrutura do sistema, assim como os elementos tabela de entradas e saídas deste.

ii) a conexão da nova estrutura de controle. Aqui, dois casos devem ser considerados:

a) o novo controle está armazenado em uma estrutura auxiliar, o que implica na realização dos seguintes passos:

- a.1) mudar os apontadores das variáveis de entrada do processo para os elementos correspondentes na nova estrutura de controle, completando assim a desconexão da estrutura de controle antiga;
- a.2) preencher os campos referentes à estrutura do novo controle no elemento raiz da estrutura do sistema;
- a.3) criar e preencher os elementos tabela de entradas e saídas do sistema;
- a.4) destruir os elementos raiz, tabela de entradas e tabela de saídas da estrutura do novo controle;
- a.5) identificar os elementos auxiliares criados na estrutura do controle e mudar os apontadores em direção dos elementos correspondentes na estrutura do processo; no final, destruir os elementos da estrutura auxiliar.

b) o novo controle está sendo declarado, o que implica na realização dos seguintes passos:

- b.1) criar um elemento auxiliar para cada entrada do processo, preenchendo-os com o nome do elemento correspondente na estrutura do processo (os demais campos são nulos);
- b.2) mudar o apontador para o elemento que representa a entrada para o correspondente elemento auxiliar, completando a desconexão da estrutura antiga;
- b.3) montar a parte da estrutura referente ao controle a partir das equações declaradas, como apresentado anteriormente; a montagem inicia pelos elementos auxiliares.

As figuras a seguir ilustram num exemplo o procedimento de substituição da estrutura de controle. A figura 7 apresenta os diagramas de blocos do sistema atual e do novo sistema a ser criado e as estruturas simplificadas do sistema atual e do novo controle a ser conectado; a figura 8 apresenta a forma da estrutura do sistema após a desconexão do controle; a figura 9 apresenta a forma da estrutura do sistema após a conexão do novo controle. Nas figuras 8 e 9, os traços fortes indicam as alterações realizadas em cada etapa, respectivamente.

#### 6.6 - Acesso aos elementos da estrutura de dados

O acesso aos elementos da estrutura é realizado de diferentes formas em função do tipo de operação desejada:

- i) a procura de elementos para a leitura ou modificação dos dados deste;





- ii) a recuperação de todos os dados armazenados na estrutura;
- iii) a recuperação das equações armazenadas para a montagem do diagrama de fluxo de sinais.

Estas diferentes formas de acesso aos elementos da estrutura, a serem descritas a seguir, estão baseadas nos métodos de atravessamento da árvore binária, os quais são apresentados resumidamente no anexo 2.

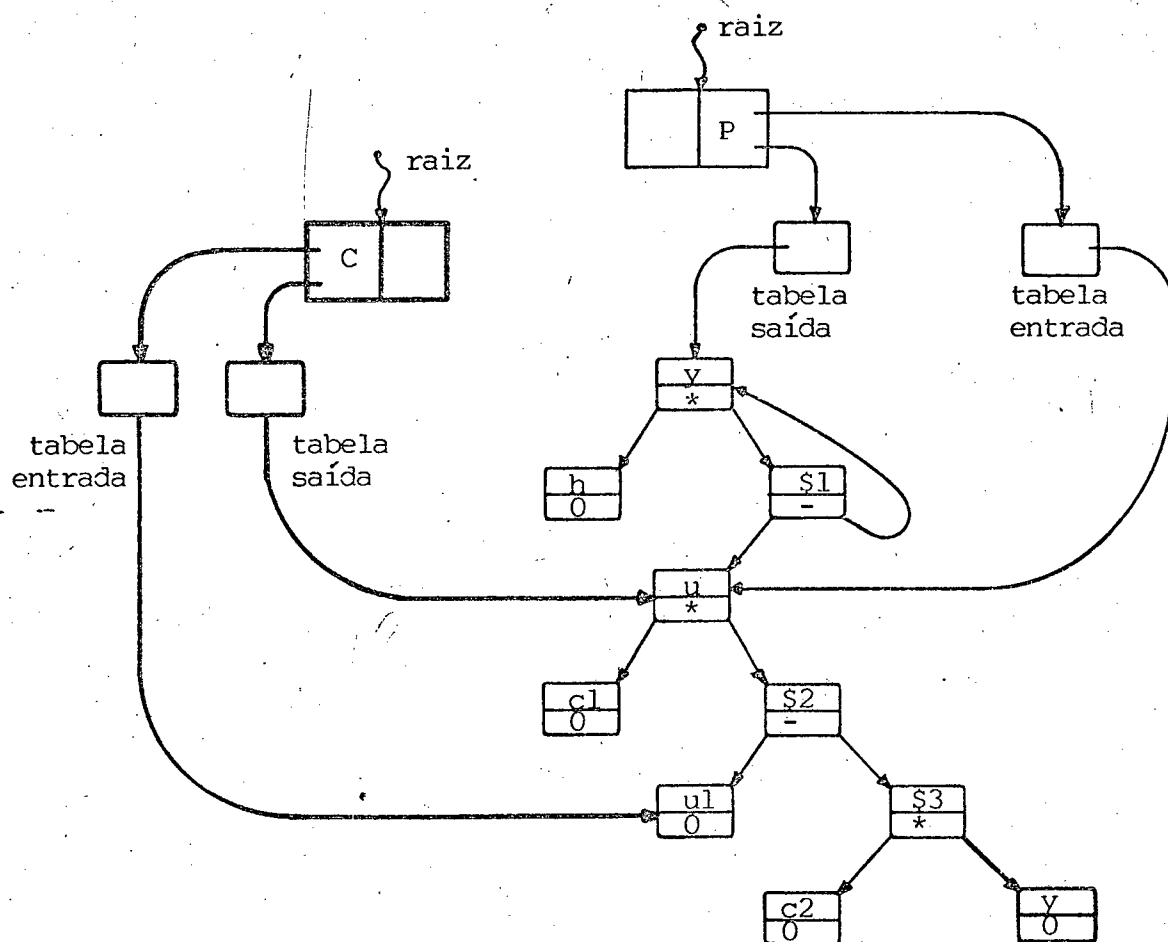


Figura 8 - Forma da estrutura do sistema após a desconexão da estrutura de controle.

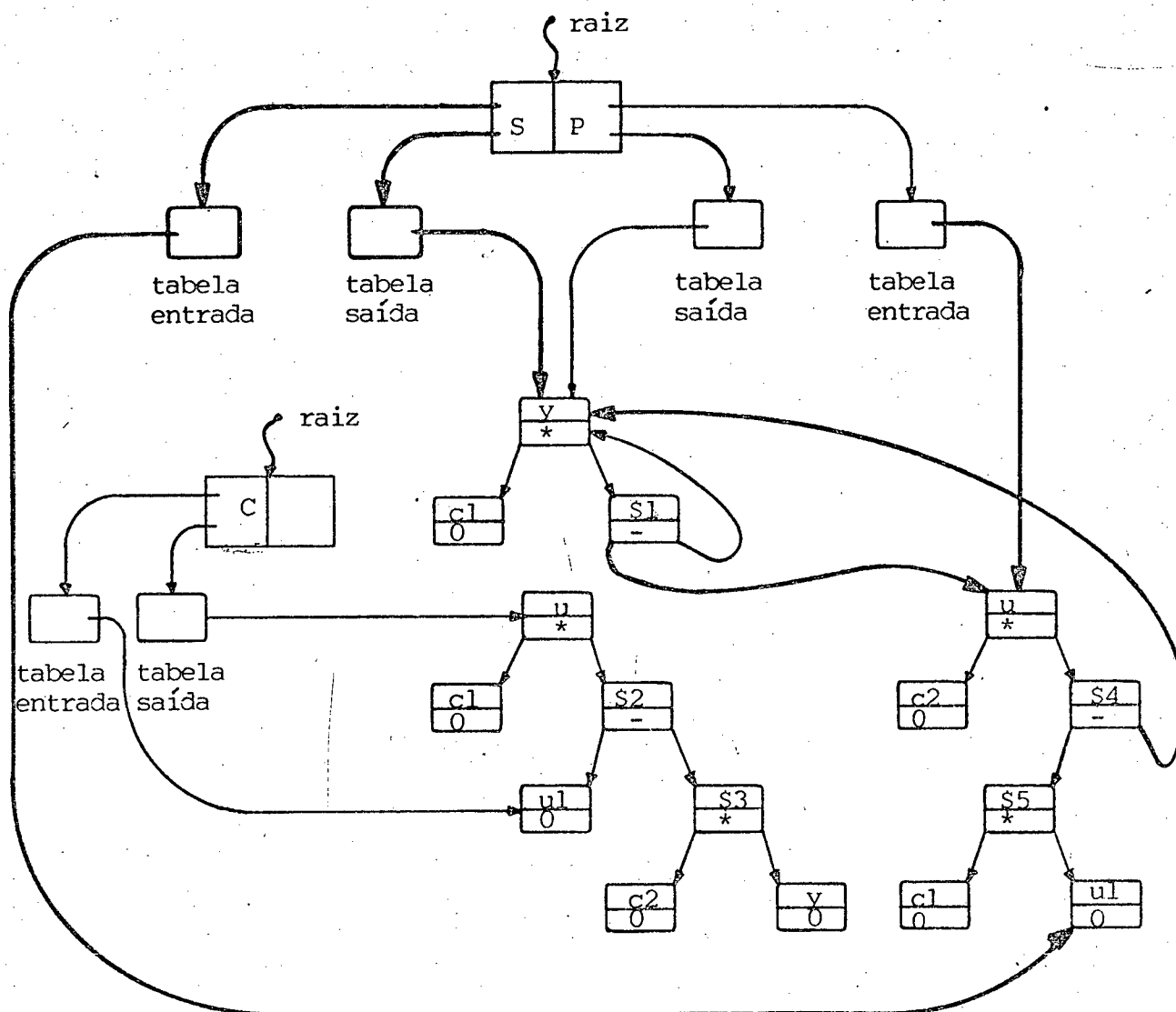


Figura 9 - Forma da estrutura do sistema após a conexão da nova estrutura de controle

#### 6.6.1 - Procura de elementos

A procura de elementos na estrutura é utilizada em vários momentos, como na montagem da árvore, na realização dos diálogos e na recuperação dos dados armazenados. Ela utiliza o atravessamento pós-ordem como forma de leitura da estrutura, sendo realizada somente para elementos que não representem variáveis de entrada, saída ou perturbação, os quais podem ser

acessados diretamente pelas tabelas associadas a raiz da estrutura.

A procura inicia por um elemento representando uma variável de saída, testando-se, antes de seguir um determinado caminho, o código associado ao apontador; o caminho é bloqueado quando atingir elementos que já apareceram na estrutura (código igual a 1 ou 2). A procura termina quando se encontrar o elemento desejado.

#### 6.6.2 - Recuperação dos dados armazenados

A recuperação dos dados armazenados na estrutura para visualização é realizada a pedido do usuário, através da diretiva informação; ela consiste em reapresentar todos os dados declarados para o processo ou um dos controles em uso. A reapresentação desses dados se faz a partir da remontagem automática do diálogo através do qual estes foram declarados, o que implica na recuperação dos dados seguindo a mesma ordem da declaração.

O processo de recuperação no caso de representação por funções de transferência está dividido nas seguintes etapas:

- i) a recuperação dos dados armazenados na raiz;
- ii) a identificação das variáveis de entrada, saída e perturbação;
- iii) a procura e identificação dos blocos e dos seus respectivos tipos;
- iv) a recuperação das equações na forma declarada através da utilização do atravessamento em ordem, com a reposição dos parênteses eliminados na transformação

das equações;

- v) a recuperação dos modelos dos blocos.

A recuperação dos dados de um processo ou controle representado por equações de estado é realizada de forma simples, visto que não é necessário procurar elementos ou recuperar equações; as informações são obtidas diretamente no único bloco que armazena o modelo.

### 6.6.3 - Recuperação das equações

A recuperação das equações, desmembradas em relações bloco a bloco, para a montagem do diagrama de fluxo de sinais é utilizada quando da realização de um cálculo formal para a obtenção da função de transferência global, tanto por solicitação do usuário como para as transformações internas de modelo necessárias em algumas ferramentas (ex: simulação).

A recuperação é realizada através do seguinte algoritmo de leitura da estrutura:

- i) iniciando a leitura por uma variável de saída, ler e armazenar o nome da variável da raiz da subárvore, o nome da variável ou do bloco da subárvore esquerda, o operador da raiz da subárvore e, finalmente, o nome da variável da subárvore esquerda.  
Caso o operador seja "+" ou "-", guardar a variável da subárvore direita em uma pilha auxiliar e seguir a partir da variável da subárvore esquerda, repetindo o processo; caso o operador seja "\*", seguir a partir da variável da subárvore direita, repetindo o processo. Em ambos os casos, verificar antes de seguir determinado caminho, o

código associado ao respectivo apontador (caso o código seja igual a 1 ou 2 o elemento apontado deve ser apenas lido, bloqueando-se o caminho);

ii) ao terminar a leitura, verificar o conteúdo da pilha auxiliar. Ler a variável do topo da pilha, procurá-la na estrutura e recomeçar a leitura a partir desta. Repetir até que a pilha esteja vazia;

iii) após esvaziar a pilha verificar se todas as variáveis de saída apareceram nas relações obtidas. Caso contrário, repetir todo o processo a partir destas.

#### 6.7 - Conclusão

Neste capítulo, foram apresentadas as estruturas para o armazenamento dos resultados, dos dados do diálogo e dos modelos do processo e do controle. Foram descritas em detalhe as características construtivas da estrutura para armazenamento dos modelos do sistema e a maneira de acessá-las para as diferentes operações necessárias ao programa proposto. A seguir, apresenta-se as principais características das ferramentas disponíveis no programa.

## CAPÍTULO 7

### DESCRIÇÃO DAS FERRAMENTAS DO PROGRAMA

Este capítulo descreve as ferramentas disponíveis no programa, apresentando as principais características da redução formal, da transformação de modelos, das ferramentas de análise e projeto clássicas e modernas e da simulação.

#### 7.1 - Redução formal

##### 7.1.1 - Introdução

Dentre os vários métodos existentes para o cálculo de transmitâncias equivalentes, o método desenvolvido por Mason (MASON, 1956) é o mais conhecido e utilizado; esse método permite calcular diretamente a transmitância equivalente  $G$  entre um nó de saída e um nó de entrada a partir da determinação dos percursos e laços do diagrama. Porém, o método de Mason não é o mais adequado para uma implementação computacional devido a ineficiência do cálculo dos laços e percursos para sistemas com grande número de nós e caminhos.

Um método mais sistemático para a obtenção da transmitância equivalente e, portanto, mais adequado para uma implementação, é o método desenvolvido por Abrahams (ABRAHAMS, 1966); esse método está baseado na montagem da matriz de transmitância do sistema e

na aplicação de uma técnica equivalente ao da eliminação gaussiana.

As modificações introduzidas por Munro (MUNRO, 1977; MUNRO e GRIFFITHS, 1985) neste método aumentaram a sua capacidade permitindo:

- i) estender o método para o caso de sistemas multivariáveis;
- ii) realizar o cálculo entre dois nós quaisquer sem que a matriz de transmitância necessite ser reconstruída para cada caso;
- iii) reduzir a matriz de transmitância nos casos em que o cálculo for realizado para apenas uma parte do diagrama.

Este foi o método escolhido para implementação no programa, sendo descrito em detalhes a seguir.

#### 7.1.2 - O método de redução formal

O método está baseado na montagem de uma matriz de transmitância aumentada  $H$ , que tem a seguinte forma:

$$H = \left[ \begin{array}{cc|cc} & T & & V \\ \hline & & & \\ \hline U & & & R \\ \hline & & & \end{array} \right] \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} n \\ \\ m \end{array}$$

$\underbrace{\hspace{10em}}_n \quad \underbrace{\hspace{5em}}_s$

onde:

$T$  : é a matriz de transmitância do sistema, de dimensões  $n \times n$  para um sistema com  $n$  nós;

$U$  : corresponde a uma matriz de dimensões  $m \times n$ , onde as  $m$  linhas correspondem aos  $m$  nós de entrada definidos;



V : corresponde a uma matriz de dimensões  $n \times s$ , onde as  $s$  colunas correspondem aos  $s$  nós de saída definidos;

R : corresponde a uma matriz de dimensões  $m \times s$  onde, após a aplicação das regras de redução, serão obtidas as transmitâncias equivalentes.

a - construção da matriz de transmitância aumentada

As  $m$  linhas e as  $s$  colunas acrescentadas para formar a matriz  $H$  tem seus elementos nulos, exceto para os elementos cuja posição corresponde aos nós de entrada e saída desejados, os quais tem valor 1. Graficamente, isso implica num diagrama de fluxo de sinais contendo nós auxiliares ligados por transmitâncias unitárias a estes nós.

b - determinação da matriz de transmitância equivalente

A matriz de transmitância equivalente é obtida a partir da aplicação das seguintes regras simples (MUNRO e GRIFFITHS, 1985):

Para  $j = 1, \dots, n$  examinar a matriz  $H$  e fazer:

i) Se  $h_{jj} = 0$ , fazer  $\Delta_j = 1$ ;  
se  $h_{jj} \neq 0$ , fazer  $\Delta_j = 1 / (1 - h_{jj})$ ;

ii) examinar os elementos da coluna  $j$  abaixo de  $h_{jj}$  (isto é, para  $i = j+1, \dots, n+m$ ) e para cada elemento  $h_{ij} \neq 0$  encontrado fazer:

ii.1) examinar os elementos da linha  $j$  a direita de  $h_{jj}$  (isto é, para  $k = j+1, \dots, n+1$ ) e para cada elemento  $h_{jk} \neq 0$  encontrado fazer:

$$h_{ik} = h_{ik} + \Delta_j * h_{ij} * h_{jk}.$$

Nos casos onde se deseja obter a transmitância equivalente de apenas uma parte do diagrama, pode-se reduzir a aritmética computacional envolvida no cálculo, eliminando-se da matriz H os nós que não estão diretamente envolvidos com o cálculo desta transmitância. A determinação dos nós que podem ser retirados, é realizada a partir da matriz de alcançabilidade R, a qual mostra, para cada nó, os nós que este pode alcançar e os nós que o alcançam. A matriz de alcançabilidade é determinada através dos seguintes passos (MUNRO e GRIFFITHS, 1985):

i) montar a matriz de adjacência A. Essa matriz é idêntica à matriz de transmitância T com os elementos não nulos substituídos por 1, isto é,  $A_{ij} = 1$  se  $T_{ij} \neq 0$ ;

ii) determinar a matriz de alcançabilidade R:

$$R = (I_n + A)^{(p-1)*}$$

onde:

$I_n$  : matriz identidade de dimensões nxn;

p : é o menor inteiro positivo tal que :

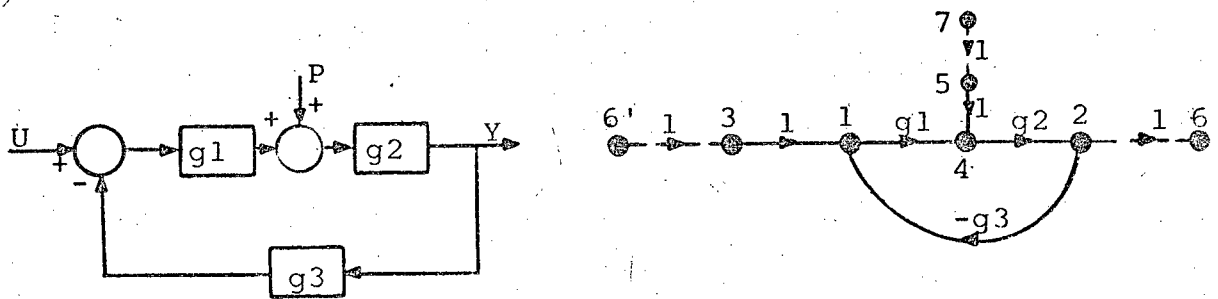
$$(I_n + A)^{(p-1)*} = (I_n + A)^p ;$$

\* : indica aritmética booleana.

A determinação dos nós que podem ser retirados para um determinado cálculo é feita examinando-se a linha e a coluna da matriz R correspondentes aos nós de entrada e saída desejados: a presença de um zero indica que o respectivo nó pode ser removido.

c - exemplo

Considere o sistema multivariável apresentado na figura 1, para o qual o diagrama de fluxo de sinais foi construído enumerando-se os nós de forma arbitrária, porém sequencial, e os nós ligados através de linhas tracejadas indicam os pontos de entrada e saída desejados.



a) diagrama de blocos

b) diagrama de fluxo de sinais

Figura 1 - Exemplo de um sistema multivariável

Para este sistema, têm-se as seguintes matrizes de transmitância \$T\$ e de transmitância aumentada \$H\$, onde "." representa zeros:

$$T = \begin{bmatrix} \cdot & \cdot & \cdot & g_1 & \cdot \\ -g_3 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & g_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot \end{bmatrix}$$

$$H = \begin{bmatrix} \cdot & \cdot & \cdot & g_1 & \cdot & \cdot \\ -g_3 & \cdot & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & g_2 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \end{bmatrix}$$

Para o sistema apresentado anteriormente na figura 1 a matriz de adjacência \$A\$ é a seguinte:

$$A = \begin{bmatrix} \cdot & \cdot & \cdot & 1 & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot \end{bmatrix}$$

A partir dela se obtém a seguinte matriz de alcançabilidade R, com  $p = 3$ :

$$R = \begin{bmatrix} 1 & 1 & \cdot & 1 & \cdot \\ 1 & 1 & \cdot & 1 & \cdot \\ 1 & 1 & 1 & 1 & \cdot \\ 1 & 1 & \cdot & 1 & \cdot \\ 1 & 1 & \cdot & 1 & 1 \end{bmatrix}$$

Através da matriz R verifica-se que, por exemplo, para o cálculo da transmitância equivalente entre Y e P (nós 2 e 5, respectivamente) pode-se eliminar a linha e a coluna 3 da matriz H, pois existe um zero na linha 5 correspondendo ao nó 3.

Para o cálculo das transmitâncias equivalentes entre Y e U (nós 2 e 3, respectivamente) e Y e P (nós 2 e 5, respectivamente) verifica-se que nenhuma linha e coluna podem ser eliminadas da matriz H, já que nenhum zero foi encontrado na coluna 2 e não existe zero comum nas linhas 3 e 5 da matriz R. Neste caso, aplicando-se as regras propostas para o método chega-se, ao final, com a seguinte matriz H:

$$H = \left[ \begin{array}{cccc|cc} \cdot & \cdot & \cdot & g_1 & \cdot & \cdot \\ -g_3 & \cdot & \cdot & -g_1 g_3 & \cdot & 1 \\ 1 & \cdot & \cdot & g_1 & \cdot & \cdot \\ \cdot & g_2 & \cdot & -g_1 g_2 g_3 & \cdot & g_2 \\ \cdot & \cdot & \cdot & 1 & \cdot & g_2 / (1 + g_1 g_2 g_3) \\ \cdot & \cdot & 1 & g_1 & \cdot & g_1 g_2 / (1 + g_1 g_2 g_3) \\ \cdot & \cdot & \cdot & \cdot & 1 & g_2 / (1 + g_1 g_2 g_3) \end{array} \right]$$

Obtendo-se, portanto:

$$Y / U = g_1 g_2 / (1 + g_1 g_2 g_3)$$

$$Y / P = g_2 / (1 + g_1 g_2 g_3).$$

## 7.2 - Transformação de modelos

A transformação de modelos somente é realizada no caso de processos monovariáveis lineares que não possuam perturbação. As possibilidades de transformação dependem da representação utilizada na declaração do modelo do processo, sendo as seguintes:

- de representação por funções de transferência contínuas (S) para representação por equações de estado contínuas;
- de representação por funções de transferência discretas (Z) para representação por equações de estado discretas;
- de representação por equações de estado contínuas para representação por função de transferência contínua (S);
- de representação por equações de estado discretas para representação por função de transferência discreta (Z);
- de representação contínua para representação discreta (discretização);

## 7.3 - Ferramentas clássicas

As ferramentas clássicas são utilizáveis somente no caso de sistemas monovariáveis, englobando os métodos do lugar das raízes

e da resposta freqüencial. Essas ferramentas serão implementadas a partir de algoritmos clássicos, cujas características são apresentadas a seguir.

### 7.3.1 - Lugar das raízes

Para este cálculo é obtido, primeiramente, o polinômio característico da função de transferência de malha fechada do sistema. Um algoritmo baseado no método de Newton-Raphson, aplicado a este polinômio, permite calcular as raízes para cada valor do parâmetro de cálculo dentro da faixa especificada pelo usuário. A escolha do parâmetro é feita pelo usuário dentre os que foram declarados no controle ou automaticamente quando não houver parâmetro declarado. As raízes calculadas são armazenadas para posterior ordenamento, com o objetivo de permitir um traçado gráfico correto.

Um novo método de cálculo do lugar das raízes, baseado no método da continuidade (BIEN e LEE, 1986), está sendo testado.

O cálculo do lugar das raízes somente é realizado para sistemas lineares, podendo ser utilizado tanto para sistemas contínuos como para sistemas discretos; no caso de sistemas híbridos, o processo é discretizado obtendo-se o lugar das raízes discreto.

### 7.3.2 - Resposta freqüencial

Este cálculo é realizado a partir da função de transferência de malha aberta na forma fatorada, obtida diretamente ou por transformação automática. A seguir, é calculado para cada valor da frequência dentro da faixa definida pelo usuário, a

contribuição de todas as singularidades, tanto em termos de ganho como de fase, e são armazenados os seguintes dados:

- logarítmo da frequência;
- módulo do ganho em dB;
- fase em graus;
- parte real do ganho;
- parte imaginária do ganho;

Esses dados permitem a construção imediata dos diagramas de Bode, Nichols e Nyquist.

#### 7.4 - Ferramentas Modernas

As estruturas de controle moderno são montadas a partir de um processo representado como na figura 2 abaixo, onde:

- U : vetor de entradas;
- $Y_c$  : vetor de saídas a controlar;
- W : vetor de perturbações;
- X : vetor de estados mensuráveis.

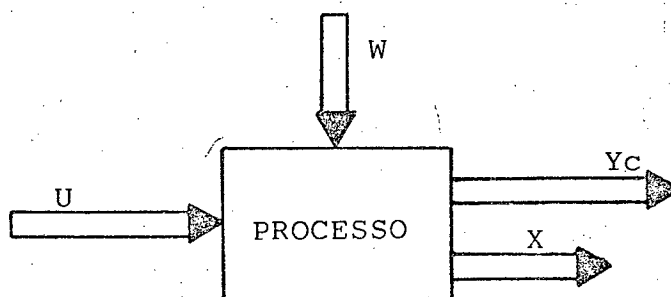


Figura 2 - Estrutura do processo

Nesta etapa do desenvolvimento do programa, as ferramentas modernas disponíveis limitam-se aos seguintes casos:

- a) controle por realimentação de estados (reais (X) ou observados (Z)):

Esta ferramenta permite analisar uma estrutura do tipo mostrada na figura 3, sendo selecionada através das leis de controle seguintes:

$$i) \quad U = Y_r + K X;$$

$$ii) \quad U = Y_r + K Z.$$

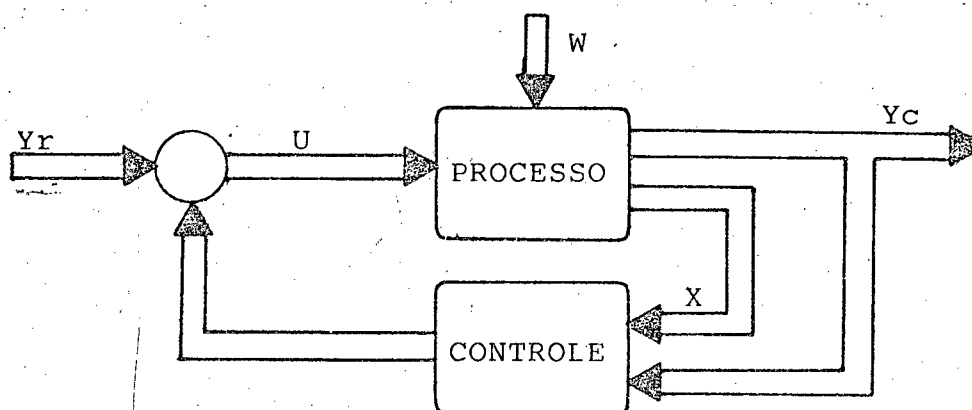


figura 3 - Estrutura para a realimentação de estados

O bloco de controle representa, neste caso, a matriz ganho de realimentação de estados com ou sem o observador.

b) controle por realimentação de estados (reais (X) ou observados (Z)) com integrador dirigido pelo erro de seguimento ( $Y_r - Y_c$ ):

Esta ferramenta permite analisar uma estrutura do tipo mostrada na figura 4, sendo selecionada através das leis de controle seguintes:

$$i) \quad U = P \int_0^t (Y_r - Y_c) dt + K X;$$

$$ii) \quad U = P \int_0^t (Y_r - Y_c) dt + K Z.$$

onde "P" representa uma matriz de ganho.



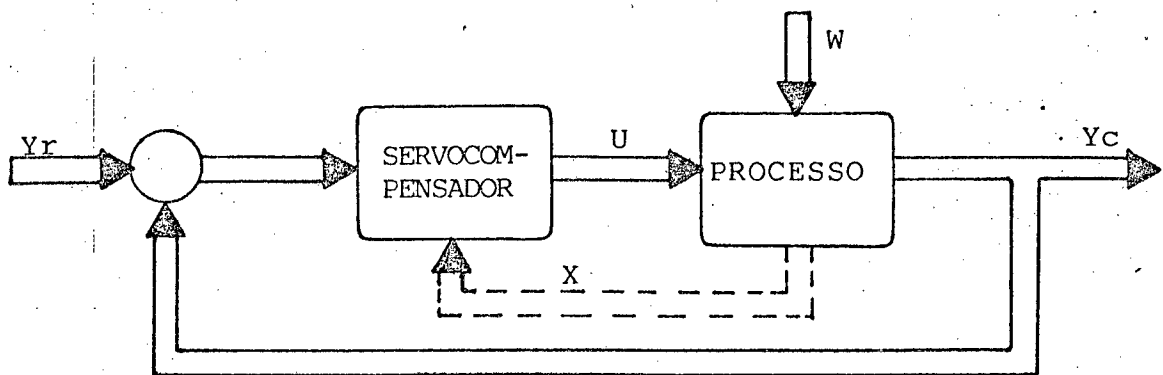


figura 4 - Estrutura para a realimentação de estados com integração do erro de seguimento

O servocompensador representa, na sua estrutura, todos os parâmetros da lei escolhida (matriz ganho de realimentação de estados, observador, etc). O servocompensador é robusto para sinais constantes, ou seja, permite que a saída siga referências constantes independente da presença de perturbações constantes, se estas forem consideradas no modelo do processo; para os demais tipos de sinais existe erro de seguimento (RODRIGUES, 1981).

Com o objetivo de unificar internamente as estruturas anteriores, a figura 3 pode ser redesenhada na forma apresentada na figura 5, com o "servocompensador" representando na sua estrutura a matriz ganho de realimentação de estados e o observador, se for o caso. Embora não se tenha aqui nenhuma ação dirigida pelo erro de seguimento, o termo servocompensador foi utilizado para manter a nomenclatura anterior. A equação geral do servocompensador, a ser apresentada posteriormente, é, neste caso, simplificada de forma a representar apenas uma realimentação de estados.

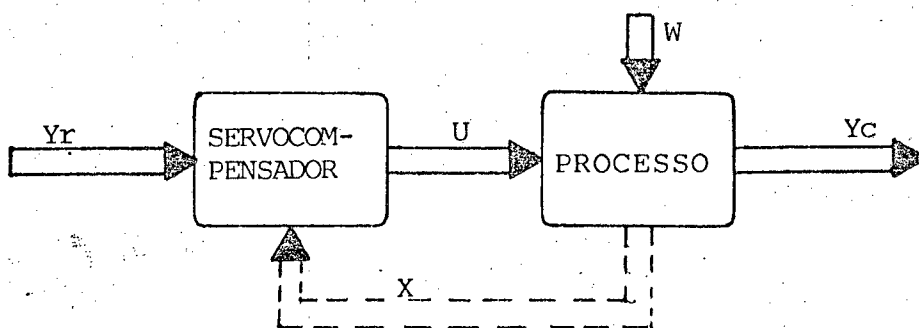


Figura 5 - Estrutura para a realimentação de estados modificada

Após a escolha da lei de controle e a definição dos dados necessários aos cálculos pelo usuário, é automaticamente montada a estrutura correspondente conforme apresentado nas figuras acima (4 e 5) e determinadas as matrizes do controle e do observador (quando necessário); todas essas operações são realizadas na diretiva de declaração do controle.

No cálculo da matriz ganho da realimentação de estados e do observador, verifica-se, respectivamente, a controlabilidade e observabilidade do processo, ou do processo aumentado, caso se considere as perturbações.

#### 7.4.1 - Definição da estrutura do servocompensador

Seja um processo representado na sua forma mais geral como:

$$\begin{aligned}\dot{X} &= A X + B U + D W \\ Y_c &= C X + E U + F W\end{aligned}\quad (1)$$

onde  $Y_c$  é o vetor de saídas a controlar. As saídas mensuráveis (saídas a controlar e estados mensuráveis) são dados por:

$$Y_m = C_m X + E_m U + F_m W \quad (2)$$

e,

$$Y_m = \begin{bmatrix} Y_c \\ X \end{bmatrix} \quad (3)$$

se os estados são acessíveis, ou:

$$Y_m = Y_c \quad (4)$$

se os estados não são acessíveis.

Considera-se que a perturbação pertence à classe modelada por equações autônomas do tipo:

$$\begin{aligned} \dot{X}_w &= A_w X_w \\ W &= C_w X_w \end{aligned} \quad (5)$$

Deseja-se representar as várias leis de controle definidas anteriormente sob a forma de um servocompensador dado por:

$$\begin{aligned} \dot{Z} &= A_c Z + B_c Y_r + D_c Y_m \\ U &= C_c Z + E_c Y_r + F_c Y_m \end{aligned} \quad (6)$$

onde:

$X \in R^n$  : vetor estados do processo;

$U \in R^p$  : vetor controle;

$Y_c \in R^p$  : vetor saídas a controlar;

$X_w \in R^s$  : vetor estados da perturbação;

$W \in R^r$  : vetor perturbações;

$Y_m \in R^q$  : vetor saídas mensuráveis;

$Y_r \in R^p$  : vetor entradas de referência;

e,

$A, B, C, D, E, F, C_m, E_m, F_m, A_w, C_w$  : matrizes de dimensões apropriadas.

#### 7.4.2 - Obtenção das matrizes do servocompensador

A tabela 8 abaixo mostra como devem ser compostas as matrizes do servocompensador para cada lei de controle.

LEIS SCOMP.	$U=Y_r+KX$	$U=Y_r+KZ$	$U=P \int_0^t (Y_r-Y_c)dt + KX$	$U=P \int_0^t (Y_r-Y_c)dt + KZ$
$A_c$	0	$A_1 - LC_1 + B_1K$	0	$\left[ \begin{array}{c c} A_1 - LC_1 + B_1K & B_1P \\ \hline 0 & 0 \end{array} \right]$
$B_c$	0	$B_1$	I	$\left[ \begin{array}{c} 0 \\ \hline I \end{array} \right]$
$D_c$	0	L	$\left[ -I \mid 0 \right]$	$\left[ \begin{array}{c} L \\ \hline -I \end{array} \right]$
$C_c$	0	K	P	$\left[ K \mid P \right]$
$E_c$	I	I	0	0
$F_c$	$\left[ 0 \mid K \right]$	0	$\left[ 0 \mid K \right]$	0

Tabela 8: Matrizes do servocompensador

A tabela 9, a seguir, mostra quais matrizes  $A_1, B_1, C_1$  devem ser usadas para o preenchimento da tabela anterior.

matrizes processo	$A_1$	$B_1$	$C_1$
$\dot{X} = AX + BU$ $Y_C = CX + EU$	$A$ $n \times n$	$(B - L E)$ $n \times p$	$C$ $p \times n$
$\dot{X} = AX + BU + DW$ $Y_C = CX + EU$	$A_a = \begin{bmatrix} A & D & Cw \\ 0 & I & Aw \end{bmatrix}$ $(n+s) \times (n+s)$	$\begin{pmatrix} (Ba - La E) \\ B \\ 0 \end{pmatrix} \begin{bmatrix} I_n \\ Ls \end{bmatrix}$ $(n+s) \times p$	$C_a = \begin{bmatrix} C & 0 \end{bmatrix}$ $p \times (n+s)$
$\dot{X} = AX + BU + DW$ $Y_C = CX + EU + FW$	$A_a$ $(n+s) \times (n+s)$	$(Ba - La E)$ $(n+s) \times p$	$C_b = \begin{bmatrix} C & F & Cw \end{bmatrix}$ $p \times (n+s)$
$\dot{X} = AX + BU$ $Y_C = CX + EU + FW$	$A_b = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}$ $(n+s) \times (n+s)$	$(Ba - La E)$ $(n+s) \times p$	$C_b$ $p \times (n+s)$

Tabela 9: Matrizes para a tabela 1

O cálculo da matriz de realimentação de estados (K), pela propriedade da separação (CHEN, 1984), é realizado sempre considerando-se todos os estados do processo acessíveis. O cálculo da matriz do observador pode ser realizado de duas maneiras:

- considerando-se apenas os estados do processo;
- considerando-se um vetor de estados aumentado dos "s" estados da perturbação.

Apresenta-se a seguir exemplos ilustrando a forma como dois elementos da tabela.8 foram obtidos.

#### Exemplo 1

Para o processo representado por:

$$\begin{aligned} \dot{X} &= AX + BU \\ Y_C &= Cx \end{aligned} \quad (7)$$

deseja-se representar na estrutura do servocompensador a seguinte lei de controle:

$$U = Y_r + KX \quad (8)$$

De (3), temos:

$$X = \begin{bmatrix} 0 & I \end{bmatrix} Y_m \quad (9)$$

Substituindo-se (9) em (8) e comparando-se o resultado com (6) obtém-se as seguintes matrizes para o servocompensador:

$$A_c = B_c = D_c = 0$$

$$C_c = 0; \quad E_c = I; \quad F_c = \begin{bmatrix} 0 & K \end{bmatrix}$$

## Exemplo 2

Para o processo representado por:

$$\dot{X} = AX + BU + DW \quad (10)$$

$$Y_c = CX + EU + FW$$

deseja-se representar na estrutura do servocompensador a seguinte lei de controle:

$$U = P \int_0^t (Y_r - Y_c) dt + K Z \quad (11)$$

agregando-se (5) e (10) obtém-se:

$$\begin{bmatrix} \dot{X} \\ \dot{X}_w \end{bmatrix} = \begin{bmatrix} A & D C_w \\ 0 & A_w \end{bmatrix} \begin{bmatrix} X \\ X_w \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U \quad (12)$$

$$Y_c = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_w \end{bmatrix} + E U$$

que pode ser escrito, fazendo-se  $X_a = \begin{bmatrix} X \\ X_w \end{bmatrix}$ , como:

$$\begin{aligned} \dot{X}_a &= A_a X_a + B_a U \\ Y_c &= C_a X_a + E U \end{aligned} \quad (13)$$

com  $A_a$ ,  $B_a$  e  $C_a$  definidas como em (12).

Definindo-se:

$$\dot{Z}_p = Y_r - Y_c \quad (14)$$

e substituindo-se em (11), obtém-se:

$$U = P \dot{Z}_p + K Z \quad (15)$$

Considerando-se o par  $(A_a, C_a)$  observável e utilizando-se um observador assintótico (CHEN, 1984), tem-se a seguinte equação para o observador:

$$\dot{Z}_a = (A_a - L_a C_a) Z_a + L_a Y_c + (B_a - L_a E) U \quad (16)$$

onde

$$Z_a = \begin{bmatrix} Z_n \\ Z_s \end{bmatrix} \quad (17)$$

observa os estados do processo,  $Z_n$ , e da perturbação,  $Z_s$ , e a matriz ganho de realimentação do observador  $L$  sendo, consequentemente, aumentada para:

$$L_a = \begin{bmatrix} L_n \\ L_s \end{bmatrix} \quad (18)$$

Substituindo-se (15) em (16), obtém-se:

$$\begin{aligned} \dot{Z}_a = & (A_a - L_a C_a + (B_a - L_a E) K_a) Z_a + \\ & + ((B_a - L_a E) P) Z_p + L_a Y_c \end{aligned} \quad (19)$$

com a matriz de realimentação de estados  $K$  sendo aumentada para:

$$K_a = \begin{bmatrix} K & 0 \\ p_{xn} & p_{xs} \end{bmatrix} \quad (20)$$

afim de compatibilizar as dimensões. A submatriz de zeros ( $p_{xs}$ ) acrescentada à matriz  $K$ , multiplica os "s" estados não controláveis correspondentes à perturbação.

Assim, (15) pode ser reescrita como:

$$U = P Z_p + K_a Z_a \quad (21)$$

A partir de (21), (19), (14) e considerando (4), pode-se escrever:

$$\begin{aligned} \begin{bmatrix} \dot{Z}_a \\ \dot{Z}_p \end{bmatrix} &= \begin{bmatrix} A_a - L_a C_a + (B_a - L_a E) K_a & (B_a - L_a E) P \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Z_a \\ Z_p \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} Y_r + \begin{bmatrix} L_a \\ -I \end{bmatrix} Y_m \\ U &= \begin{bmatrix} K_a & P \end{bmatrix} \begin{bmatrix} Z_a \\ Z_p \end{bmatrix} \end{aligned} \quad (22)$$

Fazendo-se em (22),  $Z_{sc} = \begin{bmatrix} Z_a \\ Z_p \end{bmatrix}$  e comparando-se o resultado com

(6), obtém-se as seguintes matrizes para o servocompensador:

$$A_c = \begin{bmatrix} A_a - L_a C_a + (B_a - L_a E) K_a & (B_a - L_a E) P \\ 0 & 0 \end{bmatrix}$$



$$B_C = \begin{bmatrix} 0 \\ \vdots \\ I \end{bmatrix} ; \quad D_C = \begin{bmatrix} La \\ \vdots \\ -I \end{bmatrix} ; \quad C_C = [Ka \mid P]$$

$$E_C = F_C = 0.$$

### 7.5 - Simulação

A forma de realização da simulação depende das características do sistema a ser simulado, podendo ser de três tipos: global, passo a passo e híbrida. Os métodos de integração utilizados na simulação podem ser escolhidos dentre os seguintes:

- Runge-Kutta de quarta ordem (HENRICI, 1964);
- Adams-Moulton preditor corretor (HENRICI, 1964);
- matriz de transição (CHEN, 1970).

#### a - Simulação global

Este tipo de simulação é realizado a partir da representação do processo, ou sistema, reduzida a um único bloco na forma de variáveis de estado; a simulação global permite obter, portanto, somente resultados correspondentes às variáveis de saída. Ela é realizada em dois casos:

- quando o processo, ou sistema, for declarado por equações de estado;
- quando for possível realizar a redução do diagrama de blocos.

#### b - Simulação passo a passo

Este tipo de simulação é realizada nos seguintes casos:

- quando não for possível realizar uma simulação global devido à presença de não-linearidades;

- quando se desejar obter, além das saídas, as variáveis intermediárias.

A simulação passo a passo calcula a saída de cada bloco do sistema, sendo realizada a partir da representação destes por equações de estado, caso estes sejam lineares, ou da sua tabela de definição, caso sejam não-lineares. Considera-se, para efeito de simulação, que os blocos com transferência direta possuam um pequeno atraso, assumindo-se com isso o erro decorrente; estes casos deverão ser estudados posteriormente afim de determinar uma melhor solução.

O algoritmo de cálculo passo a passo (LE BARON, 1979) é realizado diretamente sobre a estrutura de dados, fazendo as seguintes considerações:

- no instante inicial " $t_0$ " todas as variáveis são conhecidas;
- a posição de cada variável na estrutura de dados é conhecida;
- deve-se ter acesso permanente aos valores das variáveis a dois instantes consecutivos;

O algoritmo basicamente se resume em:

- i) pesquisar, na estrutura, o percurso mais longo entre o nó de saída e o nó de entrada, armazenando as variáveis na ordem em que estas apareceram;
- ii) elaborar uma tabela "variável", classificando todas as variáveis na ordem em que estas devem ser calculadas a cada passo de integração;
- iii) no caso de sistemas multivariáveis, repetir o passo i

atualizando a tabela "variável" em seguida, para todas as combinações saídas/entradas.

#### c - Simulação híbrida

Esta forma de simulação utiliza o algoritmo passo a passo descrito anteriormente, sendo as saídas dos blocos de controle atualizadas somente nos tempos correspondentes aos períodos de amostragem, ficando constantes entre estes.

#### 7.6 - Conclusão

Este capítulo apresentou informações relativas às ferramentas do programa, iniciando pela descrição do método de redução formal e das possibilidades de transformações de modelos. Em seguida, definiu-se as características do cálculo da resposta freqüencial e do lugar das raízes. Foram apresentadas, então, as características das ferramentas modernas, destacando as possibilidades atualmente oferecidas e as estruturas criadas internamente para representar estas ferramentas.

Cabe aqui ressaltar que a implementação das leis de controle definidas está sendo tratada dentro de outro trabalho de pesquisa, objeto de dissertação de mestrado em andamento, que visa estudar numericamente e implementar algoritmos de controle moderno. Isto, além de permitir -aumentar o conjunto de ferramentas disponíveis, auxiliará na definição das dimensões dos sistemas a serem tratados no programa.

Finalmente, apresentou-se as características da simulação descrevendo as suas várias possibilidades (simulação global, passo a passo e híbrida).

## CONCLUSÃO

## C A P Í T U L O 8

### CONCLUSÃO

O presente trabalho foi realizado dentro da perspectiva de desenvolvimento de ferramentas de apoio para a análise e o projeto em engenharia de controle, podendo ser utilizadas tanto a nível industrial como a nível de formação; nele foram apresentados aspectos relacionados às especificações e ao projeto de um programa de Projeto Assistido por Computador para Sistemas de Controle (PACSC).

Este programa permite colocar à disposição do usuário, ferramentas para o manuseio dos modelos e para a análise e o projeto de sistemas de controle mono e multivariáveis, contínuos e discretos.

A interação programa-usuário é realizada através de mecanismos de acesso, tais como: menus ou diretivas para a seleção de ferramentas e diálogos do tipo pergunta-resposta para a entrada dos dados necessários à utilização da ferramenta selecionada. A apresentação simultânea de diferentes informações numa tela dividida em janelas e o uso de um editor de textos orientado para a entrada de dados são facilidades complementares oferecidas ao usuário.

O programa é construído a partir de um núcleo formado por um gerenciador que controla a execução deste através do sequenciamento das tarefas a serem realizadas. O gerenciador tem acesso às demais partes do programa (interface homem-máquina e unidades funcionais), ativando os mecanismos de comunicação e de apresentação de resultados, as ferramentas e os utilitários; ele assume também a montagem e o acesso à base de dados e a coordenação da área de memória destinada a esta, alocando dinamicamente espaço quando necessário.

O programa descrito se encontra em fase de projeto e implementação em um microcomputador compatível IBM-PC-XT, utilizando a linguagem Pascal e, quando indispensável, Assembly. O cronograma prevê um prazo de um ano durante ao qual serão concluídos o projeto e a implementação dos módulos correspondentes às funções especificadas nos diagramas SADT, a integração destes a partir de uma combinação das técnicas Top-Down e Bottom-Up e a realização dos testes correspondentes.

Futuros trabalhos relacionados a este programa devem tentar diminuir as restrições impostas nas especificações apresentadas, melhorando, também, os aspectos gerais do programa. Para tanto, apresenta-se as seguintes sugestões:

- modificar a forma de declaração do modelo do sistema, possibilitando ao usuário declarar o modelo de cada bloco indistintamente por funções de transferência ou equações de estado;
- possibilitar a declaração dos diagramas correspondentes aos modelos dos sistemas de forma gráfica;
- modificar a entrada de dados permitindo, também, que as

respostas aos diálogos possam ser introduzidas previamente como argumentos da diretiva selecionada.

- ampliar as possibilidades de transformação de modelos, permitindo tratar modelos multivariáveis e os casos onde o processo possua perturbações;
- ampliar o conjunto de métodos de análise e projeto, particularmente no caso de controle moderno;
- estudar e identificar soluções para a simulação passo a passo de sistemas possuindo blocos com transferência direta;
- permitir a otimização automática de parâmetros do controle;
- introduzir uma linguagem de alto-nível orientada ao problema que permita ao usuário construir suas leis de controle e programar sua aplicação;
- introduzir conceitos heurísticos e de experiências práticas visando a orientação do usuário, pelo programa, na tomada de decisões entre as várias alternativas de projeto;

Neste trabalho constatou-se também o grande número de problemas a serem resolvidos e a necessidade de um esforço conjugado de grupos de pesquisa trabalhando na área de PACSC, para que esta nova metodologia de projeto seja colocada à disposição do engenheiro de controle brasileiro, num prazo curto, recuperando assim o atraso em relação aos outros países. Para tal, é necessário um incentivo a ações que visam trocas de experiência, a padronização de rotinas, a criação de um banco de programas e a definição e o uso de técnicas comuns de desenvolvimento dos programas.

A N E X O S



## ANEXO 1 : Structured Analysis and Design Technique - S.A.D.T.

Este anexo apresenta as principais características da metodologia SADT e faz parte de publicação interna do Laboratório de Controle e Microinformática, LCMI 86/5 (FARINES, 1986-1); maiores detalhes sobre esta metodologia são encontrados em (Ross, 1977) e (Ross, 1985); exemplos de uso de SADT são apresentados em (Rodrigues et al, 1981) e (DSCAC, 1985).

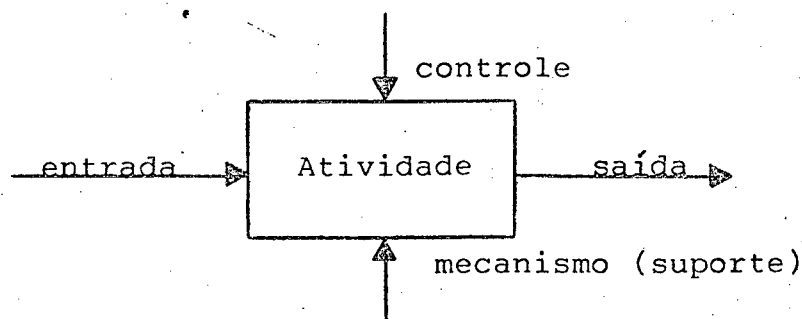
### 1 - Características Gerais da Linguagem:

Esta linguagem utiliza uma representação gráfica com caixas e setas que podem ser acopladas, representando a decomposição das atividades (atigrama) ou dos dados (datagrama). A seguir serão apresentadas as características para um atigrama por ser a decomposição utilizada neste trabalho.

A unidade básica é a caixa que representa uma atividade. As setas indicam geralmente um fluxo de dados entre atividades.

As atividades são sempre caracterizadas por um verbo e os dados por um substantivo.

A representação gráfica num atigrama é a seguinte:



### 2 - Principais convenções Utilizadas nos Diagramas:

**Seta de Entrada** : representa uma entrada utilizada pela atividade.

**Seta de Controle** : representa o controle de uma atividade

**Seta de Saída** : representa o resultado de uma atividade

**Seta de Mecanismo** : representa a realização de parte das atividades da caixa pelo "mecanismo" representado pela seta.

**Seta de Chamada** : quando a totalidade da atividade é tomada em conta pelo "mecanismo" representado pela seta, a seta é dirigida para fora da caixa; neste último caso, não é necessário decompor esta caixa.

**Dupla Seta** : ela indica uma relação de interdependência entre duas atividades, os dados fluindo nos dois sentidos. A sua representação é uma dupla seta com dois nomes separados por um traço.

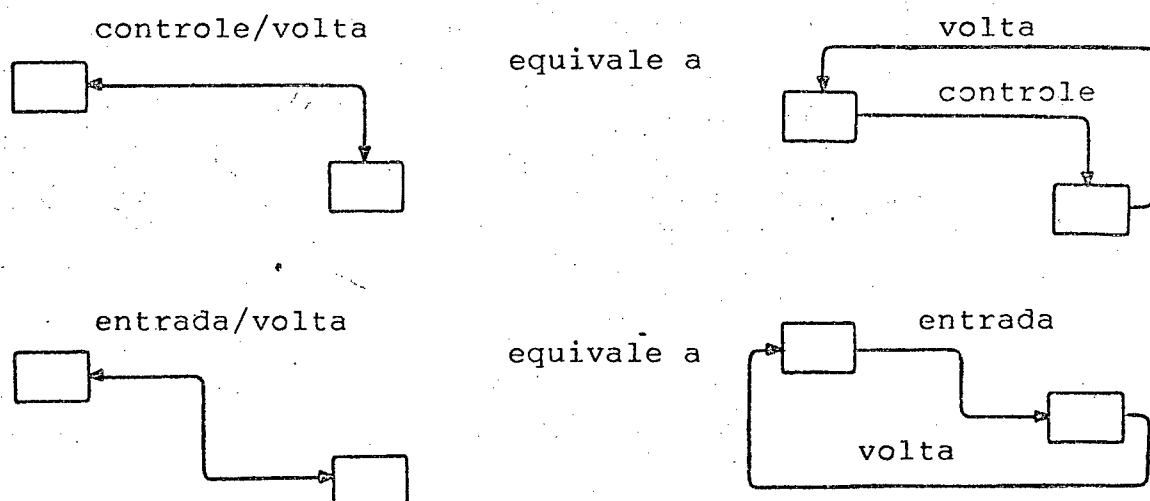


Figura 1 - Equivalência da dupla seta

### 3 - Detalhamento e Referência dos Diagramas:

A metodologia SADT segue um modelo de representação evolutivo com uma série de diagramas que permitem detalhar a representação de nível superior. Ela segue uma decomposição top-down com limitação de 6 caixas no máximo para cada nível de decomposição; cada diagrama é então criado a partir do detalhamento de cada elemento ("caixa") do diagrama de nível superior. A caixa a ser detalhada é chamada CAIXA PARENTE e o diagrama DIAGRAMA PARENTE; os diagramas resultantes do detalhamento de cada caixa são chamados DIAGRAMAS FILHOS.

A decomposição não pode ser infinita; quando se pensa ter atingido um nível de detalhamento suficiente, um complemento de informações ao diagrama pode ser redigido em qualquer linguagem, desde que este pode ser contido na própria página correspondente ao diagrama.

Como referenciar os diagramas?

A referência de cada diagrama é indicada no canto esquerdo da folha-diagrama e é chamada Nó; o nó num diagrama é identificado pela letra A seguida de um número composto da seguinte maneira: os primeiros algarismos correspondem ao número de referência do diagrama da caixa parente que ele está detalhando; o último algarismo representa o número da caixa do diagrama parente que está sendo decomposta. Cada caixa em cada diagrama tem um número de referência escrito no seu canto direito. Um exemplo da forma de referenciar é apresentado na figura 2.

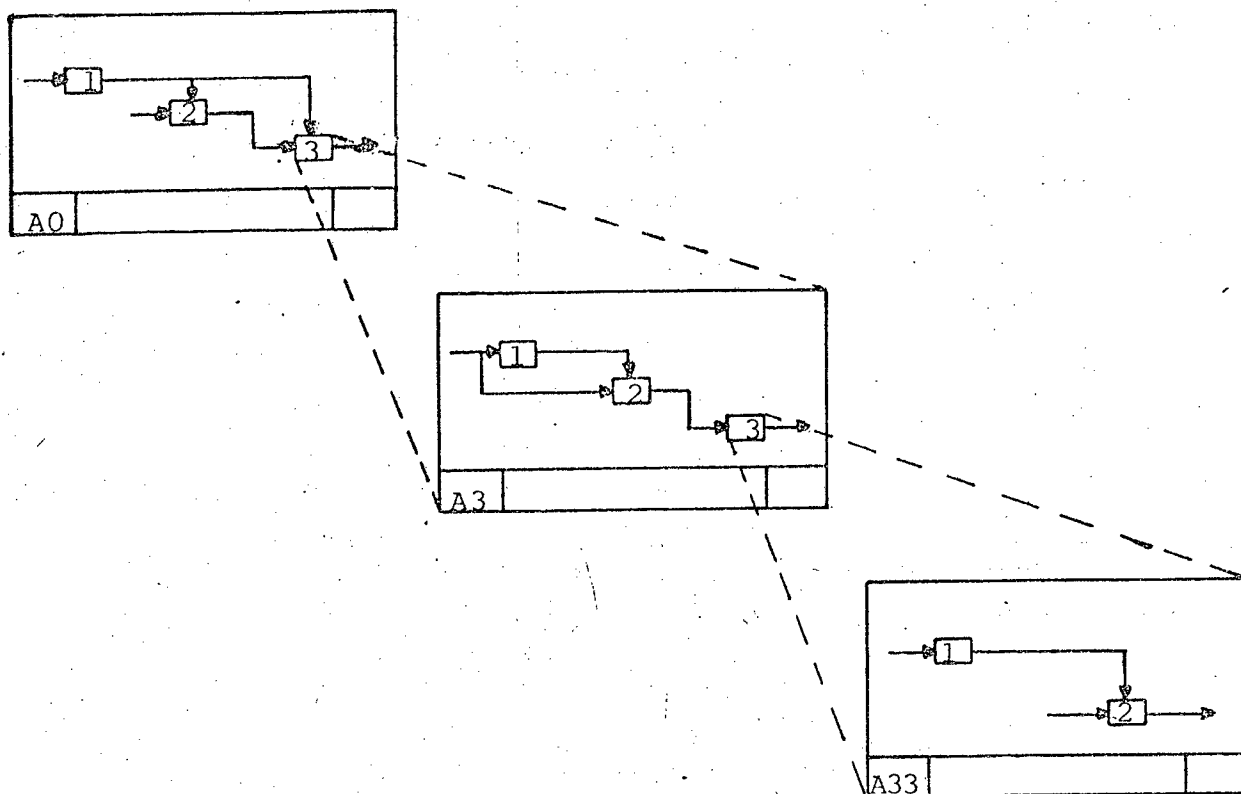
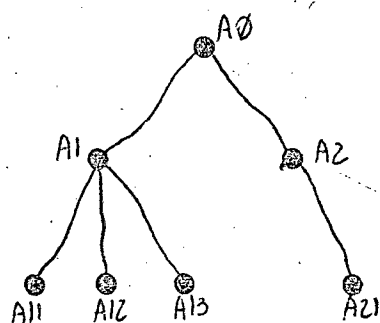


Figura 2 - Detalhamento de diagrama

A estrutura hierárquica completa do modelo do sistema pode ser representada em um diagrama chamado índice dos Nós geralmente apresentadas no início do documento final. A representação é apresentada na figura 3.



A0	Título Diagrama
A1	Título Diagrama
A11	Título Diagrama
A12	Título Diagrama
A13	Título Diagrama
A2	Título Diagrama
A21	Título Diagrama

Figura 3 - índice dos Nós

#### 4 - Etiquetagem das Setas:

Uma regra inviolável é que não pode haver adições ou

supressões durante o detalhamento: os interfaces (entrada, controle, saída, mecanismo: ICOM) e os nomes da caixa parente deverão ser os mesmos que os do diagrama filho.

As setas que tem uma extremidade livre dentro do diagrama filho indicam Entradas (I), Controle (C) e Saída (O) e Mecanismo (M) da caixa parente. Após ter verificado que todas estas setas existem no diagrama parente, elas podem ser etiquetadas com as letras I, C, O e M; a letra é seguida por um número que indica sua posição geométrica relativa. A ordem crescente de numerotação é de cima para baixo e da esquerda para a direita na caixa parente; por exemplo, uma seta "C3" no diagrama filho indica que a seta aparece como a terceira seta de controle da caixa parente. As etiquetas (I1, I2, C1) serão escritas nas extremidades não conetadas do diagrama-filho e se referenciarão a entradas, saídas, controles e mecanismos da caixa parente. Um exemplo está apresentado na figura 4.

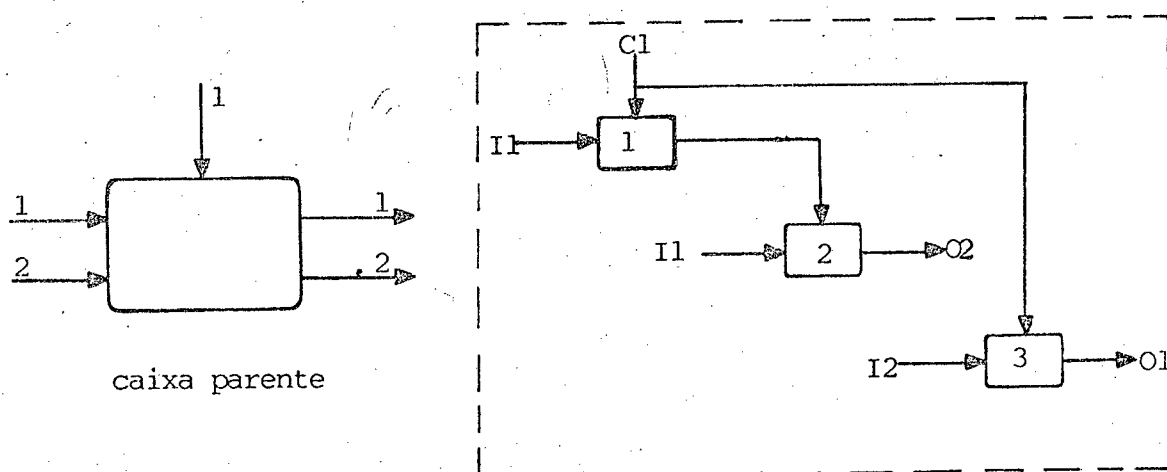


Figura 4 - Etiquetas no diagrama-filho

## 5 - Normas de Representação:

Os diagramas serão apresentados em folhas de formato normalizado conforme a figura 5.

LCMI UFSC	Nome Projeto: Autor: Data: Versão:		ETAPAS	LEITOR	DATA
			Proposta		
			Aprovação		
			Publicação		
Nó	Título				Referência

Número do  
Nó

Ex.: A41

Título da  
caixa parente

Referência do  
diagrama-filho

Ex.: Rp13

Figura 5 - Folha-diagrama

As caixas serão representadas na forma apresentada na figura 6. O número da caixa aparece em baixo a direita de cada caixa, a referência do diagrama-filho aparece em baixo do número da caixa; esta referência indica a página que contém o detalhe da caixa, a sua ausência significa que não foi feita a descrição mais detalhada.

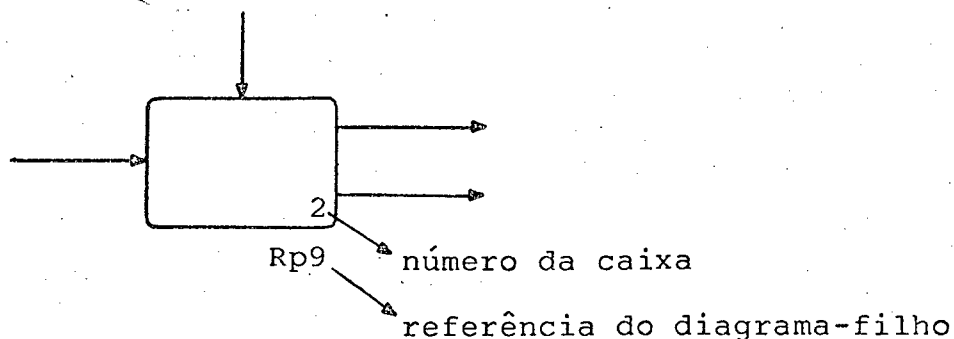


Figura 6. - Representação de caixa

## ANEXO 2 : Principais tipos de estruturas de dados

Dentre os tipos clássicos de estruturas de dados encontrados na literatura destacam-se as listas e as árvores.

### a - Listas

A lista representa o tipo de estrutura de dados mais comum e freqüentemente utilizado, podendo ser encontrada de duas formas:

- i) na forma seqüencial, na qual os elementos são colocados em posições contíguas da memória. A forma mais comum de representação das listas seqüenciais são os arranjos;
- ii) na forma linkada (encadeada), na qual os elementos são colocados em posições aleatórias da memória sendo interligados através de apontadores.

As principais vantagens das listas encadeadas sobre as listas seqüenciais estão relacionadas com a inserção e deleção de elementos em qualquer posição da lista, a qual é realizada pela simples mudança de dois apontadores. Porém, muitas linguagens de programa somente permitem o arranjo como única forma de estruturação de dados.

As duas formas mais comuns de listas são a pilha ("stack") e a fila ("queue"). A pilha é uma lista onde todas as inserções e retiradas se realizam numa extremidade, denominada topo; a fila é uma lista onde as inserções ocorrem numa extremidade, a traseira, e as retiradas na outra extremidade, a frontal.

As figuras 1 e 2 apresentam representações da pilha e da fila nas formas seqüencial e encadeada, respectivamente.

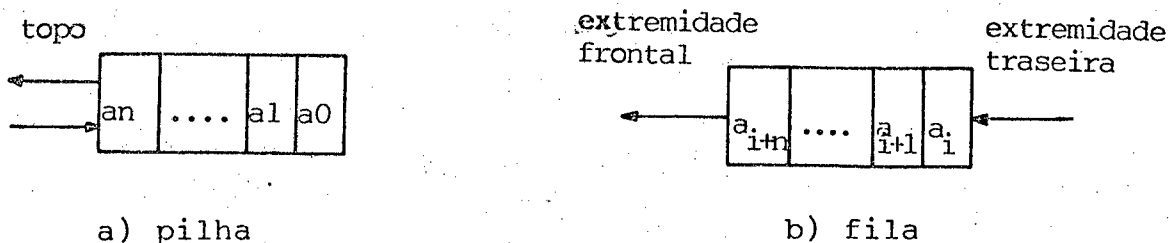


Figura 1 - Representação sequencial.

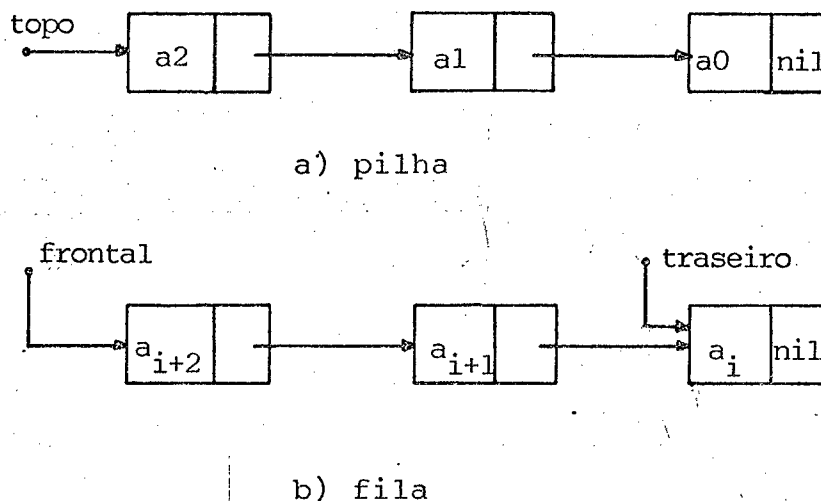


Figura 2 - Representação encadeada

Maiores detalhes sobre as listas podem ser encontrados em (HOROWTIZ e SHANI, 1984) e (KRUSE, 1984).

## b - Árvores

A árvore pode ser definida como um conjunto finito de nós (elementos), tais que:

- i) existe um nó especial denominado de raiz;
- ii) os demais nós estão desdobrados em  $n$  conjuntos disjuntos, cada um desses constituindo uma árvore, denominados de subárvores.

Este tipo de estrutura é largamente utilizada na área de computação, como por exemplo, na teoria de compiladores (árvores



de análise), na manipulação de arquivos (árvores de diretórios), em inteligência artificial (árvores de decisão), etc.. A figura 3 apresenta uma estrutura típica de uma árvore.

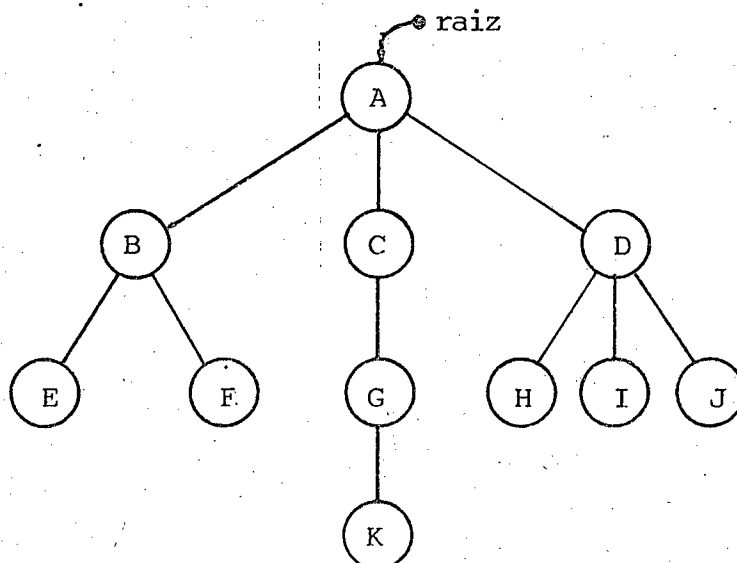


Figura 3 - Estrutura do tipo árvore.

Dentre os tipos de estrutura em árvore desenvolvidos, a árvore binária é uma das mais importantes e frequentemente utilizadas. Ela se caracteriza pelo fato de cada nó possuir no máximo duas subárvores, e pelo fato de se qualificar diferentemente uma subárvore esquerda de uma subárvore direita. A figura 4 apresenta uma estrutura típica de uma árvore binária.

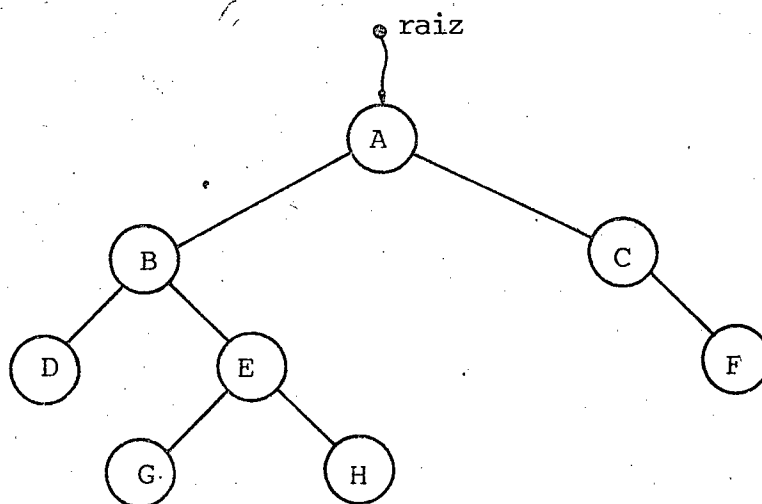


Figura 4 - Estrutura do tipo árvore binária.

## b.1 - Atravessamento da árvore binária

O atravessamento de uma árvore binária significa a leitura de cada elemento uma única vez, de modo a recuperar todas as informações armazenadas na árvore. As três formas principais de atravessamento são:

### i) atravessamento em ordem:

Esta forma de atravessamento implica em ler primeiro a subárvore esquerda, depois a raiz da subárvore e finalmente a subárvore direita. Esta operação requer descer pela árvore para a esquerda até não poder mais. Ao ler o elemento muda-se para a direita, repetindo o processo. Se não for possível mover-se para a direita voltar um elemento para trás.

### ii) atravessamento pré-ordem:

Esta forma de atravessamento implica em ler primeiro a raiz da subárvore, depois a subárvore esquerda e finalmente a subárvore direita. Esta operação requer ler o elemento e seguir pela esquerda, repetindo o processo. Quando não for possível continuar, deslocar-se para a direita e recomeçar, ou ir para trás até poder seguir para a direita, recomeçando em seguida.

### iii) atravessamento pós-ordem:

Esta forma de atravessamento implica em ler primeiro a subárvore esquerda, depois a subárvore direita e finalmente a raiz da subárvore. Esta operação requer descer pela árvore para a esquerda até não poder mais. Ao ler o elemento seguir pela direita e recomeçar, ou ir para trás até poder seguir pela

direita, recomeçando em seguida.

A árvore de expressão apresentada na figura 5 ilustra as formas de atravessamento em ordem, pré-ordem e pós-ordem, as quais produzem:

- em ordem:  $a + b / c * d - e * f$
- pré-ordem:  $* + a / b c - d * e f$
- pós-ordem:  $a b c / + d e f * - *$

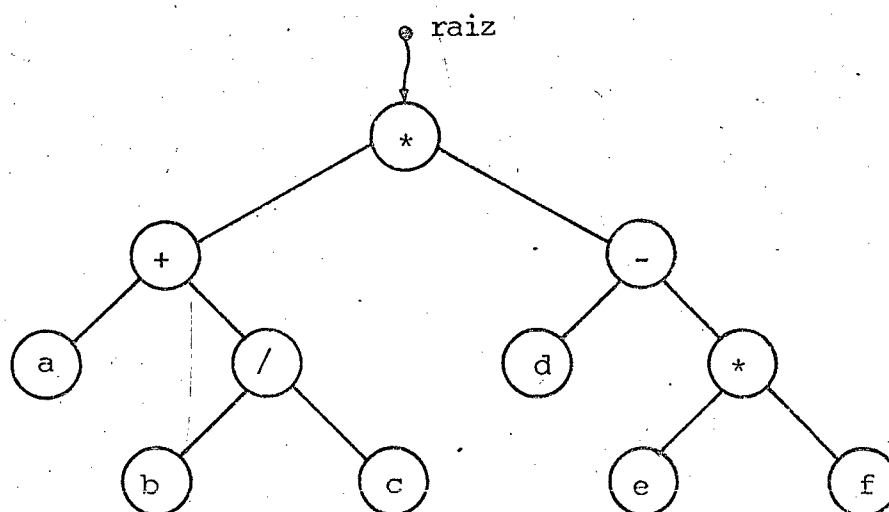


Figura 5 - Árvore binária para a expressão:  
 $(a + b / c) * (d - e * f)$ .

Informações mais detalhadas sobre as árvores podem ser encontradas em (HOROWITZ e SHANI, 1984), (KRUSE, 1984) e (WIRTH, 1976).

## BIBLIOGRAFIA

ABRAHAMS, J.R., Control, January, p.31, February, p.88; March, p.150, 1966;

ASTROM, K.J., "Computer Aided Modeling and Design of Control Systems - A Perspective", IEEE control system magazine, vol 3: 4-16, 1983;

BIEN, Z., LEE, J., "A Note on a Computer-Aided Root-Locus Method", Automatic Control, No 3, March, 1986;

BUNZ, D. e GUTCHOW, K., "CATPAC - An Interactive Software Package for Control System Design", Automatica, vol 21, No 2: 209-213, 1985;

CHEN, C.T., "Introduction to Linear System Theory", Halt, Rinehout and Winston, INC, New York, 1970;

DENHAN, M. J., "Design Issues for CACSD Systems", Proceedings of the IEEE, vol 72, No 12, December, 1984;

"Projeto DSCAC: Projeto funcional do protótipo CADLOG", Nota interna CTI-IA, vol. 1 e 2, Dezembro, 1985;

ESMANHOTO, L. M. G., "Computer Aided Design - Fundamentos e Tecnologia", tutoriais do primeiro CONAI, São Paulo, 1983;

FARINES, J.M., "Metodologia de Desenvolvimento de Sistemas", Nota interna LCMI: 86/5, Julho, 1986-1;

FARINES, J.M., "Documentação de Software", Nota interna LCMI: 86/2, Abril, 1986-2;

FARINES, J. M., SAVI, V. M. e BRUCIAPAGLIA, A. H., "Projeto Assistido por Computador para Sistemas de Controle: Um Pacote Interativo", 6º Congresso Brasileiro de Automática, Belo Horizonte, 1986;

GOMIDE, F.A.C. e SZAJNER, J., "Análise, Síntese e Projeto em Sistemas de Controle Auxiliado por Computador", Anais do Segundo Conai, p216-223, 1985;

HENRICI, P., "Elements of Numerical Analysis", John Wiley & Sons, Inc., New York, 1964;

JAMES, J.R., FREDERICK, D.K. e TAYLOR, J.H., "The Use of Expert-Systems Programming Techniques for the Design of Lead-Lag Compensators", IEE conference publication 252 - Control 85, 9th-11th, University of Cambridge, p180-185, 1985;

HOROWITZ, E. e SAHNI, S., "Fundamentos de Estruturas de Dados", Editora Campus Ltda, Rio de Janeiro, 1984;

JAMSHIDI, M. e HERGET, C.J., "Computer-Aided Control System Engineering", North-Holland, 1985;

KRUSE, R.L., "Data Structures & Program Design", Prentice-Hall, New Jersey, 1984;

LE BARON, J.P., "Conception Assistee en Automatique", These de Docteur-Ingenieur, Universite de Rennes, France, 1979;

LITTLE, J.N., EMAMI-NAEINI, A. e BANGERT, S.N. , "CTRL-C and Matrix Environments for the Computer-Aided Design of Control Systems", em Jamshidi e Herget: "Computer-Aided control System Engineering", North-Holland, 1985;

MASON, S.J., Proceedings Inst. Radio Engineers, 44, 920, 1956;

MUNRO, N., "Composite System Studies Using the Connection Matrix", Int. Journal of Control, vol 26, No 6, p. 831-839, 1977;

MUNRO, N. e GRIFFITHS, M., "Graphical Input of System Descriptions", IEE conference publication 252 - Control 85, 9th-11th, University of Cambridge, p. 192-197, 1985;

PINE, R.A., WALKER, R.A. e GUPTA, N.K., "MATRIXx/PC: Modeling, Simulation and Optimization on Personal Computers", Proceedings of the IEEE, vol 73, No 12, 1985;

PRESSMAN, R.S., "Software Engineering: A Practitioner's Approach", McGraw-Hill, 1982;

QUENC'HOU, Y. e BUISSON, J., "Logiciel de Simulation et de Conception Assistée pour Processus Continus", RGE - Revue Générale de l'Electricite, No 1, p27-33, 1985;

RODRIGUES, A. R. S., "Proposição de uma Metodologia Sistematizada para o Projeto de Controladores de Carga Freqüência", Dissertação de Mestrado - EEL, UFSC, Junho, 1981;

RODRIGUES, P., NOYES, D. e BEOUNES, C., "Projet d'étude du Système de Commande et de Surveillance d'une machine flexible de production", Note interne LAAS 81-I-09, Mars, 1981;

ROSS, D.T., "Structured Analysis (SA): A Language for Communicating Ideas", IEEE Transaction on Software Engineering, vol SE-3, No 1, January, 1977;

ROSS, D.T., "Applications and Extensions of SADT", Computer, vol. 18, No 4, April, 1985;

ROSEMBROCK, H.H., "Computer-Aided Control System Design", Academic Press, London, 1974;

SCHMID, C., "A Workstation for Computer Aided Analysis and Design of Control Systems", IEE conference publication 252 - Control 85, 9th-11th, University of Cambridge, p489-494, 1985;

SPANG III, H.A., "The Federated Computer-Aided Control Design System", Proceedings of the IEEE, vol 72, No 12, December, 1984;

TAYLOR, J.H. e FREDERICK, D.K., "An Expert System Architecture for Computer-Aided Control Engineering", Proceedings of the IEEE, vol 72, No 12, December, 1984;

TOZZI, C.L., "PAC - Projeto Auxiliado por Computador", Editora da UNICAMP/editora Papirus, Campinas, 1986;

WIRTH, N., "Algorithms + Data Structures = Programs", Prentice-Hall, New Jersey, 1976;

YOURDON, E. and CONSTANTINE, L.L., "Structured Design", Yourdon Press, New York, 1978;

ZENI, N., FERNANDES, N.C., STEMMER, M., CASTELAN, E.B., HAPPEL, F.P., BARBOSA, F.P., FARINES, J.M., "Ensino Assistido por Computador: Análise e Projeto de Sistemas de Controle por Microcomputador", Anais do quinto CBA: 395-401, 1984.